BRSA-ESDS: A Binary Reptile Search Algorithm for Extractive Single Document Summarization

Abdelaali Bekhouche, Mohamed Boussalem, and Hicham Haouassi

Original scientific article

***Abstract*—Automated summarization systems are becoming more popular due to the growing volume of text information in many real-life applications. This paper presents a novel approach to summarising a single document by modeling it as an optimization problem and using the Reptile Search Algorithm (RSA) to solve it. This algorithm is inspired by crocodile hunting behaviour, which includes two main steps encircling and hunting. The encircling step requires high walking or belly walking phases while the hunting step requires coordination or cooperation. In this study, we propose a binary version of this algorithm called BRSA-ESDS to implement an automatic text summarization system by choosing a subset of the sentences in the original text. This algorithm optimizes an objective function to preserve linguistic quality based on many factors, including readability and consistency in the compressed summary while improving its coverage. This model ensures the diversity and coverage of selected sentences in the summary by optimizing a harmonic average of the objective function factors. Additionally, this model controls the summary's length to ensure its readability. The results are compared with state-of-the-art approaches using ROUGE measures on the Document Understanding Conference (DUC) benchmark datasets. According to ROUGE scores, our approach consistently performs better than other methods.**

***Index Terms*—Extractive text summarization, reptile search algorithm, multi-objective optimization, meta-heuristic algorithm, swarm intelligence.**

I. INTRODUCTION

Nowadays, with an increasing amount of unstructured textual content available on the internet, it is essential to use effective text summarization systems. These systems are crucial for quickly extracting valuable insights, especially when dealing with redundant or ambiguous information that often hides the intended meaning. For example, finding specific details in scientific articles can be time-consuming. A well-designed summarizer can significantly improve text clarity, readability, and efficiency by reducing the time spent searching for relevant information.

Text summarization in natural language processing mainly follows two approaches: extractive and abstractive summarization. Abstractive summarization generates a concise summary while paraphrasing the original content. In contrast, extractive summarization ranks and selects the most important sentences from the source document [1]. Summarization techniques can also be divided into single-document and multi-document summarization. Single-document summarization processes one text source, while multi-document summarization fuses information from multiple related sources, such as news clustering systems.

The summarization problem has been widely studied and explored using different machine learning paradigms, including traditional machine learning [2] and deep learning approaches [3]. Recently, researchers have investigated meta-heuristic optimization techniques to solve extractive summarization tasks [4]. These algorithms select relevant sentences based on linguistic and statistical criteria. Evolutionary and swarm intelligence algorithms have shown significant improvements over existing methods [5], [6].

A recent addition to swarm intelligence is the Reptile Search Algorithm (RSA), introduced by Abualigah et al. [7]. RSA simulates crocodile hunting strategies consisting of two main behavioral phases: encircling the prey and cooperative attack. This method has demonstrated promising performance in continuous optimization problems. Motivated by its efficiency, we adapt RSA to handle extractive summarization, which is formulated as a binary optimization problem.

In this work, we introduce a binary version of RSA, named BRSA-ESDS, to build an automatic extractive text summarization system that selects a subset of sentences from the input document. The proposed model evaluates multiple linguistic factors to ensure readability, coverage, and consistency in the generated summary while maintaining an adequate length.

The main contributions of this paper are summarized as follows:

- We propose a novel binary adaptation of the Reptile Search Algorithm that preserves its exploration–exploitation mechanism in a discrete sentence-selection space, rather than relying on conventional transfer-function-based binarization.
- We introduce a sentence-level binary encoding and update strategy that enables RSA to operate on combinatorial extractive summarization problems effectively.
- We design a harmonic multi-objective fitness function that jointly optimizes relevance, redundancy reduction, and summary length constraints within a unified optimization framework.
- We develop BRSA-ESDS, an extractive single-document summarization system, and validate its effectiveness on DUC benchmark datasets using ROUGE metrics, achieving competitive performance compared to state-of-the-art methods.

To clarify the technical novelty of the proposed approach, it

Manuscript received November 6, 2025; revised January 12, 2026. Date of publication March 31, 2026. Date of current version March 31, 2026.

Authors are with the ICOSI Laboratory, Abbes Laghrour University, Khenchela, Algeria (e-mails: {bakhouche.abdelali, boussalem_mohamed, haouassi.hichem}@univ-khenchela.dz).

Digital Object Identifier (DOI): 10.24138/jcomss-2025-0187

is essential to note that BRSA-ESDS is not a straightforward binarization of the original Reptile Search Algorithm. Unlike existing binary metaheuristic approaches that primarily rely on generic transfer functions to map continuous positions into binary decisions, BRSA-ESDS introduces a problem-specific binary adaptation tailored to sentence selection in extractive summarization. The proposed sentence-level encoding and update strategy preserves the cooperative hunting behavior and exploration–exploitation balance of RSA within a discrete optimization space. Furthermore, the proposed harmonic multi-objective fitness function explicitly integrates relevance, redundancy reduction, and summary length constraints, enabling RSA to address the combinatorial nature of extractive single-document summarization effectively.

The rest of this article is organized as follows. Section II reviews related works. Section III introduces the original Reptile Search Algorithm and highlights the motivation behind its binary adaptation. Section IV presents the proposed binary formulation. Section V reports and discusses the experimental results. Section VI concludes the paper and outlines possible future directions.

II. Related Works

Automatic text summarization methods are generally categorized into supervised approaches, unsupervised or deep learning-based methods, and metaheuristic-based extractive summarization. The strengths and limitations of these techniques are summarized below.

A. Supervised and Transformer-Based Approaches

Supervised extractive summarization relies on labeled datasets for training. Lamsiyah et al.[10] used sentence embeddings with a feed-forward neural network to learn sentence relevance, while Parikh et al. [11] introduced a weakly supervised model for summary quality estimation. Bao et al. [12] extended the idea using mutation-based augmentation strategies.

A more recent trend focuses on transformer architectures. Afsharizadeh and Mousavirad[13] provided a comprehensive survey of extractive transformer-based methods, highlighting improved semantic representation yet noting computational inefficiency for long documents. Dan et al. [14] combined extractive and abstractive processing based on Lawformer for legal summarization, improving ROUGE scores but requiring significant memory resources. Although supervised models demonstrate strong performance, they depend on costly data annotation and extensive training resources.

B. Unsupervised and Graph-Based Approaches

Unsupervised summarization avoids labeled data. Srivastava et al. [15] combined clustering with topic modeling for extractive summaries, while Isonuma et al. [16] built discourse trees for unsupervised abstractive summarization. Song et al. [17] proposed the FEOM clustering-based approach.

More recently, Umair et al. [18] introduced a heterogeneous GNN model integrating CNN/BiLSTM encoding with enhanced LDA topic modeling, enabling long document processing but increasing overall complexity and runtime.

These approaches ensure thematic coverage but may lack readability or grammatical fluency.

C. Metaheuristic-Based Extractive Summarization

Since sentence selection is a combinatorial optimization task,metaheuristic algorithms have been widely investigated to maximize coverage, reduce redundancy, and maintain coherence. Verma and Om [19] optimized feature weights through metaheuristics, while Debnath et al. [20] used modified Cat Swarm Optimization for multi-objective summarization. Multi-agent optimization was explored by Mirshojaee et al. [21]. For multi-document summarization, Verma et al.[22] employed shark smell optimization, and Rautray and Balabantaray [23] adopted Cat Swarm Optimization with TF-IDF and cosine similarity. Other bio-inspired techniques include Social Spider Optimization [24]. A recent review confirmed a continuous shift toward optimization-based summarization for low-resource conditions while noting limitations regarding readability constraints and adaptivity [25]. Similarly, Siranjeevi et al. [26] hybridized BPSO and Masked GA to improve convergence stability but required careful parameter tuning. Discussion: Neural models excel in semantic understanding whereas metaheuristics efficiently search large solution spaces. A persisting research gap exists in approaches that jointly enforce readability, diversity, coverage, and summary length control. Motivated by these findings, we propose a binary reptile search algorithm that integrates these objectives to generate compact and informative summaries.

III. Preliminaries

A. Overview of RSA

The reptile search algorithm is a new metaheuristic algorithm developed by Abualigah and colleagues in 2022 [2]. It is inspired by the hunting behavior of crocodiles. This section describes how the algorithm was inspired by real-life crocodile behavior and adapted to computational needs.

B. The Natural Origins of RSA

Recent research suggests that crocodiles can hunt in groups, demonstrating high levels of intelligence and coordination. These creatures have slow metabolisms and tend to hunt and eat mainly at night in shallow water using ambush tactics. When they gather in a group, they corral fish into a small space using a bait ball and then take turns catching the fish. Crocodiles of different sizes have various roles in the hunting process, with larger crocodiles luring fish into shallow lagoons and smaller, more intelligent crocodiles preventing the fish from escaping. Sometimes, crocodiles startle other animals, like zebras or pigs, into fleeing into a lagoon where other crocodiles wait. After the hunt, the crocodiles migrate out of the area before regrouping and appearing to relax. [27].

C. Artificial RSA

The primary behavior of crocodiles involves encircling and hunting their prey. The Reptile Search Algorithm (RSA) is based on these behaviors, with the exploration (encircling) and exploitation (hunting) phases being mathematically modeled to drive the optimization process. Like other swarm algorithms, RSA starts by setting up a group of potential solutions with random positions. These solutions are then assessed for their fitness, and their positions are adjusted through exploration and exploitation phases to approach the optimal solution in each iteration.

1) Encircling Phase (exploration): Once a crocodile has located potential prey, it transitions from the exploration phase to a more focused approach. It begins to stalk its prey slowly and deliberately, often partially submerged in water, to get within striking distance. This behaviour minimizes the chance of alerting the prey and maximizes the crocodile's chance of a successful attack. Crocodiles exhibit two types of movement during encircling: high walking and belly walking [28]. High walking is a slow, stalking movement that crocodiles use when searching for prey or exploring their environment. Belly walking is a faster movement when a crocodile is chasing prey or trying to escape danger. In the context of the RSA, these movements represent the algorithm's ability to explore the search space of possible solutions [29]. The RSA mirrors this stalking behaviour mathematically to exploit the promising areas of the search space where potential optimal solutions are likely to be found. It works as follows:

- Identifying the target: The algorithm identifies the best solution in the population. This solution represents the "prey" that the other solutions (crocodiles) will try to encircle.
- Movement towards the prey: The positions of the other solutions are updated iteratively, moving them closer to the best solution. This movement is guided by a mathematical formula that considers:
  - The current position of each solution.
  - The position of the best solution.
  - Randomization factors causing slight movement variations prevent local optimal solutions.
- Adaptive step size: The step size determines how far each solution moves towards the best solution and is dynamically adjusted throughout the optimisation process. In the initial stages, more significant steps are taken to explore the search space effectively. As the algorithm progresses and converges towards the optimal region, the step size is gradually reduced to fine-tune the solutions.

The encircling phase offers several advantages: Firstly, it enables the algorithm to efficiently exploit promising areas by concentrating on the vicinity of the best solution, facilitating the discovery of better alternatives nearby. Secondly, it fosters convergence as the encircling behaviour gradually steers the population towards the optimal or near-optimal solution. Thirdly, it maintains a balance with the exploration phase, ensuring the algorithm doesn't become trapped in local optima and retains the capacity to explore diverse regions within the search space. The RSA can switch between two phases: exploration (encircling) and exploitation (hunting). This transition is governed by four conditions, which are determined by dividing the total number of iterations into four segments. During the exploration phase, RSA employs two primary search strategies: high walking and belly walking. These strategies are used to probe the search regions and strive to discover an improved solution. When $k \leq \frac{K_{max}}{4}$, high walking is utilized, while belly walking is used when $\frac{K_{max}}{4} < k \leq \frac{K_{max}}{2}$. As a result, around 50% of the exploration steps involve high walking, and the remaining 50% involve belly walking. Additionally, a random scaling factor affects each element, increasing the variety of solutions and enabling the exploration of various areas within the search space. This method uses a simple yet efficient process that imitates the adaptive behavior of crocodiles in their natural habitat. The following formulas are suggested for adjusting the position during the exploration phase.

if $k \leq K_{max}/4$.

$$z_{(p,q)}(k+1) = Opt_q(k)\left(-\lambda_{(p,q)}(k)\right)\gamma - S_{(p,q)}(k)\,\text{rnd} \quad (1)$$

if $K_{max}/4 < k \leq K_{max}/2$.

$$z_{(p,q)}(k+1) = Opt_q(k)\, z_{(r_2,q)}\, FE(k)\,\text{rnd} \quad (2)$$

where $Opt_q(k)$ is the best solution obtained at the q^{th} position, K_{max} and k are the maximum and current iteration numbers respectively, rnd is a random number in the range $[0, 1]$, $\lambda_{(p,q)}(k)$ is the value of the hunting operator of the p^{th} solution at the q^{th} position which is calculated as shown in equation (2), γ is a crucial parameter that determines the precision of the exploration process, while $S_{(p,q)}(k)$ denotes the reduction function, which constrains the search space and is computed according to equation (3), and $FE(k)$, referred to as Final Exploration, is a probabilistic ratio that takes values between $[-2, 2]$. It can be expressed mathematically as in equation (4).

$$\lambda_{(p,q)}(k) = Opt_q(k) \cdot P_{(p,q)} \quad (3)$$

$$S_{(p,q)} = \frac{Opt_q(k) - z_{(r2,q)}}{Opt_q(k) + \epsilon}. \quad (4)$$

$$FE(k) = 2 \cdot r_3 \cdot \left(1 - \frac{1}{K_{max}}\right). \quad (5)$$

The symbol ϵ represents a small value. The variables r_2 and r_2 denote random numbers chosen from the ranges $[1, N]$ and $[-1, 1]$ respectively. The variable r_2 serves as a correlation coefficient. The term $P(p, q)$ denotes the percentage deviation between the q^{th} position of the best solution found so far and the current solution, as defined in equation (5).

$$P_{(p,q)} = \gamma + \frac{z_{(p,q)} - S(z_p)}{Opt_q(k) \cdot \left(UB_{(q)} - LB_{(q)}\right) + \epsilon}, \quad (6)$$

where γ is a sensitive parameter fixed at 0.1, $S(z_p)$ is the p^{th} solution's average position calculated using equation (6), $UB_{(q)}$ and $LB_{(q)}$ are the upper bound and lower bound of the q^{th} position.

$$S(z_p) = \frac{1}{n}\sum_{q=1}^{n} z_{(p,q)} \tag{7}$$

2) Hunting Phase (exploitation) : This stage is inspired by the crocodile's hunting mechanism. It is divided into hunting coordination and cooperation. Hunting coordination refers to how crocodiles work together to surround and capture their prey and cooperation refers to how crocodiles share their prey once it has been captured. In the context of the RSA, these behaviours represent the algorithm's ability to exploit the best solutions found during the exploration stage.

During the hunting phase, coordination is implemented when iterations fall within the range $\frac{K_{max}}{2} < k \leq \frac{3}{4}K_{max}$, while cooperation is implemented when iterations fall within the range $\frac{3}{4}K_{max} < k \leq K_{max}$. Random coefficients are used to navigate the local search space to produce optimal solutions and avoid getting stuck in local optima.

if $\dfrac{K_{max}}{2} < k \leq \dfrac{3K_{max}}{4}$.

$$z_{(p,q)}(k+1) = Opt_q(k)\, P_{(p,q)}(k)\, \text{rnd} \tag{8}$$

if $\dfrac{3K_{max}}{4} < k \leq K_{max}$.

$$z_{(p,q)}(k+1) = Opt_q(k) - \lambda_{(p,q)}(k)\, \epsilon - S_{(p,q)}(k)\, \text{rnd} \tag{9}$$

where $Opt_q(k)$ represents the optimal solution found at the q^{th} position. The percentage difference $P(p,q)$ is determined by comparing the q^{th} position of the best solution found with that of the current solution, as described in equation (7). $\lambda_{(p,q)}(k)$ is the hunting operator computed using equation (2), and $S_{(p,q)}(k)$ is the reduce function calculated via equation (3).

IV. Proposed Approach

We aim to generate a concise summary from a document (Doc) by selecting k informative sentences from the set $[sent_1, sent_2, \ldots, sent_k]$. The overall process of BRSA-ESDS, including population initialization and summary generation, is illustrated in Fig. 1. To satisfy summary length constraints, the search space includes all possible sentence combinations (e.g., 2^k candidates). The most effective summaries are identified based on their fitness values, where non-dominated solutions are retained by simultaneously evaluating multiple objective functions such as coverage, readability, and redundancy control.

A. Pre-Processing

In text summarization, it is crucial to understand that we cannot directly work with raw text data. Raw text is unstructured and noisy, filled with various elements such as punctuation, capitalization, special characters, and irrelevant words, which may not contribute to understanding the text's core meaning. Therefore, a preprocessing phase is indispensable in any text summarization pipeline.

This preprocessing phase transforms raw text into a more digestible and analyzable format. These steps may include tokenization, where the text is broken down into individual words or tokens; case normalization, where all text is converted to a standard case (usually lower); removal of punctuation and special characters; and often, the removal of stop words, which are common words like 'and', 'is', 'in', that usually do not carry significant meaning.

By preprocessing the text, we are cleaning and standardizing the data, making it easier for our text summarization algorithms to focus on the important aspects of the text and generate accurate, concise summaries. Without this crucial step, our algorithms would try to find meaning in a sea of noise, leading to inefficient processing and potentially inaccurate results.

B. TF-IDF Overview

TF-IDF, a cornerstone of text mining and natural language processing, quantifies a word's significance within a document relative to a broader corpus. The TF component measures term frequency, while IDF adjusts for term prevalence across multiple documents, emphasizing less frequent, distinctive words. In extractive summarization, TF-IDF is commonly used to rank sentences based on relevance, with higher scores often indicating key concepts. Although relatively simple, TF-IDF remains a valuable tool for identifying important information. To enhance summarization performance, it can be effectively integrated with more sophisticated algorithms, such as the Binary Reptile Search Algorithm (BRSA).

$$TF(t,d) = \frac{\text{Count of term } t \text{ in document } d}{\text{Total terms in document } d} \tag{10}$$

$$IDF(t,D) = \log\left(\frac{N}{1 + |\{d \in D : t \text{ occurs in } d\}|}\right) \tag{11}$$

$$TF\text{-}IDF(t,d,D) = TF(t,d) \times IDF(t,D) \tag{12}$$

where:

- N is the total number of documents in the corpus.
- $|\{d \in D : t \text{ is present in } d\}|$ denotes the number of documents in which term t appears.

C. Population Representation

In the realm of extractive text summarization, a common challenge is to represent the essence of the original text in a summary. One approach to address this challenge is using a binary vector representation. Let's consider an original text composed of n sentences, denoted as $S = \{s_1, s_2, \ldots, s_i, \ldots, s_n\}$, where s_i is the ith sentence in the input text after tokenization. The objective of extractive summarization is to select a subset of these sentences that best captures the text's main ideas. We can define the summary as a binary vector $B = \{b_1, b_2, \ldots, b_i, \ldots, b_n\}$, where each element b_i corresponds to the sentence s_i in the original text. The value of b_i is determined by whether s_i is included in the summary. Specifically, we have:

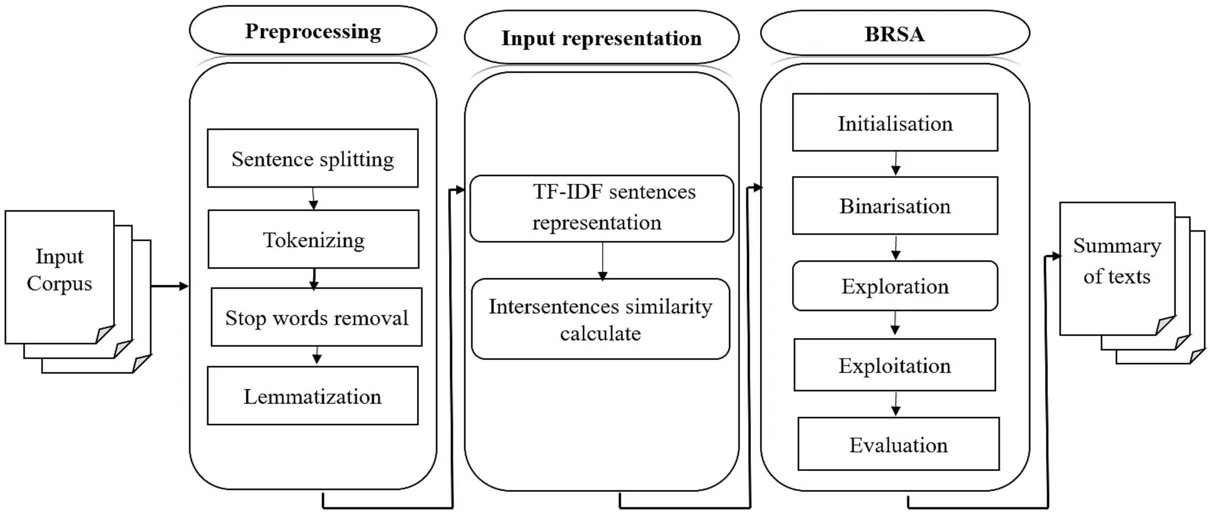


Fig. 1. Flowchart of BRSA-ESDS

$$b_i = \begin{cases} 1, & \text{if } s_i \text{ is included in the summary,} \\ 0, & \text{otherwise.} \end{cases} \quad (13)$$

For instance, consider a text with twenty sentences: $S = \{s_1, s_2, s_3, s_4, \ldots, s_{20}\}$. Suppose our extractive summary includes the text's first and last sentence. The corresponding binary vector would be $B = \{1, 0, 0, 0, 0, 0, 0, \ldots, 0, 0, 0, 1\}$.

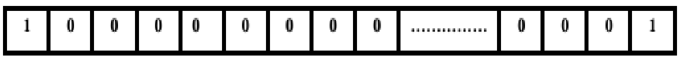


Fig. 2. Binary vector representation of sentences

The binary vector representation provides an efficient and clear summary of the original text. It helps us easily identify which sentences are included in the summary and can be used for further processing or analysis.

D. Binary Reptile Search Algorithm (BRSA)

The Binary Reptile Search Algorithm (BRSA) flowchart is presented in this section. The flowchart outlines the steps involved in RSA, including the encircling stage, which involves high walking and belly walking, and the hunting stage, which includes hunting coordination and cooperation.

1) Initialization Phase: The initialization phase is the first step in the Reptile Search Algorithm (RSA). It sets the stage for the exploration and exploitation phases. During this phase, a population of solutions (X) is randomly generated within the problem's search space bounds. Each "crocodile" represents a potential solution to the optimization problem, and a vector of decision variables represents the position of each "crocodile" in the search space. The quality of each "crocodile" (i.e., how good the solution it represents is) is evaluated using the objective function of the optimization problem. After evaluation, the best "crocodile" (i.e., the one with the highest fitness value) is identified and stored. This best "crocodile" will guide the other "crocodile" movements in the subsequent exploration and exploitation phases.

$$X = \begin{bmatrix} x_{1,1} & \cdots & x_{1,j} & \cdots & x_{1,n-1} & x_{1,n} \\ x_{2,1} & \cdots & x_{2,j} & \cdots & x_{2,n-1} & x_{2,n} \\ \vdots & \cdots & \vdots & \cdots & \vdots & \vdots \\ x_{i,1} & \cdots & x_{i,j} & \cdots & x_{i,n-1} & x_{i,n} \\ \vdots & \cdots & \vdots & \cdots & \vdots & \vdots \\ x_{N-1,1} & \cdots & x_{N-1,j} & \cdots & x_{N-1,n-1} & x_{N-1,n} \\ x_{N,1} & \cdots & x_{N,j} & \cdots & x_{N,n-1} & x_{N,n} \end{bmatrix}$$

where $x_{i,j}$ represents the j^{th} dimension in the i^{th} solution, n is the number of dimensions of the given problem, and N stands for the total number of candidate solutions. Each solution is stochastically generated using equation (12).

$$x_{i,j} = \text{rand} \times (\text{UB} - \text{LB}) + \text{LB}, \quad j = 1, 2, \ldots, n \quad (14)$$

where UB and LB are the upper and lower bounds of the given problem, and rand is a random value between $[0, 1]$.

2) Criteria for Evaluating Summary Quality: The summary quality is determined by three factors: readability, cohesion and coverage. The factors were normalized to a 0-1 range. Below are the details of these criteria:

$$\text{Coverage} = \frac{\sum_{i=1}^{n} \sum_{j=1}^{n} \text{sim}(s_i, s_j)}{N-1} \quad (15)$$

The outer summation iterates over each sentence s_i in the summary, with i ranging from 1 to n, where n is the length of the summary. The inner summation traverses all sentences s_j in the original text, with j ranging from 1 to N, where N is the length of the original text. The function $\text{sim}(s_i, s_j)$ quantifies the similarity between sentence s_i from the summary and sentence s_j from the original text. This function can be any metric that measures similarity, such as cosine similarity between their vector representations. The sum of these similarity measures is then normalized by dividing by $(N-1)$, which adjusts the overall similarity score by accounting for the length of the original text minus one.

The resulting value is the coverage score. A higher coverage score means the summary includes a larger proportion of the information in the original text.

$$\text{Cohesion} = \frac{\sum_{i,j=1}^{n}(1-\text{sim}(s_i,s_j))}{\left(n * \left(\frac{n-1}{2}\right)\right)} \tag{16}$$

The sum runs over all pairs of sentences s_i and s_j in the summary, where i and j range from 1 to n, the number of sentences in the summary. $\text{sim}(s_i, s_j)$ is the similarity function between sentences s_i and s_j. $1-\text{sim}(s_i, s_j)$ is the dissimilarity between the sentences, so the sum essentially adds up all the dissimilarities between pairs of sentences in the summary. The result of the sum is then divided by $\left(n \times \frac{n-1}{2}\right)$, which is the total number of unique pairs of sentences in the summary. The resulting value is the cohesion score. A lower cohesion score indicates that the sentences in the summary are more similar to each other and, therefore, the summary is more cohesive.

$$\text{Readability} = \frac{\sum_{i,j=1}^{n}\text{sim}(s_i,s_j)}{(n-1)} \tag{17}$$

The resulting value is the readability score. A higher readability score means the sentences in the summary are more similar to each other and, therefore, the summary is more readable.

3) Fitness Function: The objective function evaluates the quality of a candidate summary by jointly considering three complementary criteria: coverage, cohesion, and readability, as defined in Equations (15), (16), and (17), respectively. Each criterion is normalized to the range $[0, 1]$ to ensure numerical comparability and to prevent any single metric from dominating the optimization process due to scale differences.

Let V, C, and R denote the normalized coverage, cohesion, and readability scores of a candidate summary, respectively. The overall fitness value is computed as a weighted aggregation of these criteria, expressed as follows:

$$\text{Fitness} = \alpha \times V + \beta \times C + \gamma \times R \tag{18}$$

where α, β, and γ are non-negative weighting coefficients such that $\alpha + \beta + \gamma = 1$.

The weights α, β, and γ control the relative importance of coverage, cohesion, and readability in the optimization process. In this study, the three weights are set to equal values ($\alpha = \beta = \gamma = \frac{1}{3}$), reflecting an unbiased treatment of the three criteria in the absence of task-specific prior preferences. This choice ensures that the optimization process promotes balanced summaries that simultaneously preserve informative content, maintain sentence-level coherence, and achieve acceptable linguistic readability.

The use of a weighted aggregation allows the proposed BRSA-ESDS framework to incorporate multiple quality dimensions while remaining computationally efficient and flexible. A higher fitness value indicates a better-quality summary according to the combined evaluation criteria and guides the binary Reptile Search Algorithm toward optimal sentence selection.

4) Position Updating (binarization): In this work, each potential solution is represented as a binary vector within the population. However, a challenge arises when updating the position using equations (1) and (7) because these functions produce continuous values which cannot be directly represented in a binary vector. To solve this problem, we employed the sigmoid function. The sigmoid function, recognized for its characteristic 'S'-shaped curve, effectively transforms a continuous input into a value between 0 and 1. This characteristic makes it ideal for generating a binary representation from a continuous value. Thus, by applying the sigmoid function to the output of the position updating functions, I maintained the binary vector representation of solutions while incorporating the continuous updates from equations (17) and (18).

$$X_{sig} = \frac{1}{1+e^{-x}} \tag{19}$$

$$X_b = \begin{cases} 1, & \text{if } Random \leq X_{sig} \\ 0, & \text{if } Random > X_{sig} \end{cases} \tag{20}$$

The binary representation of the real-valued solution vector is denoted by X_b, while $Random$ is a random threshold value for the sigmoid function. The sigmoid function is used in the initialization phase because the upper and lower bounds of the binary space are set to 1 and 0, respectively. The position of the initialized particle is determined by the "rand" value, which is a continuous value that we can't use in a binary space. To address this, we use the sigmoid function to set the population's initial position.

5) Binary Reptile Search Algorithm (BRSA) Pseudo-code: We propose the Binary Reptile Search Algorithm (BRSA) to address the optimization challenges in binary spaces. The following pseudo-code details the steps of the BRSA, illustrating how it initializes parameters, evaluates the fitness of solutions, and iteratively updates the population to converge towards an optimal solution. This method leverages a combination of sigmoid functions and dynamic parameter updates to navigate the binary search space efficiently.

The Binary Reptile Search Algorithm (BRSA) presented in the pseudo-code aims to find the optimal solution within a binary search space by iteratively refining a population of candidate solutions. The algorithm's effectiveness stems from its dynamic adaptation strategies, which adjust the exploration and exploitation balance through different phases: high walking, belly walking, hunting coordination, and hunting cooperation. The BRSA makes it easy to switch between different search strategies by using the sigmoid function to change positions in the binary space. This makes it better at getting out of local optima and converge to a global solution. The performance of BRSA will be demonstrated and evaluated in the subsequent sections through various benchmark tests and comparative analyses against other state-of-the-art algorithms.

V. Experimental Results

This section will discuss the experimental results of using the Reptile Search Algorithm for extractive text summarization. We will evaluate the algorithm's performance by testing different particle sizes to see how they affect the quality of the resulting summaries.

Algorithm 1 Pseudo-code (BRSA)

Get input text
Pre-Processing text
Initialize parameters α, β, N, n, T
Initialize population randomly with eq. (12). $X_i : i = 1, \ldots, N$. In the binary space using sigmoid function eq. (18)
Represent each individual of the population in a vector of N cells.
while $(t \leq T)$ **do**
    Calculate the fitness value for each individual using eq (16).
    Find the optimal solution until now
    Update the Evolutionary sense ES using equation (4)
    for $i = 1$ to N **do**
        for $j = 1$ to N **do**
            Use equations (2), (3) and (5) to update the parameters η, R, and P.
            if $(t \leq \frac{T}{4})$ **then**
                Use equation (1) for High walking
                Update position using the sigmoid function eq. (18)
            else if $(t \leq \frac{T}{2}$ and $t > \frac{T}{4})$ **then**
                Use equation (1) for Belly walking
                Update position using the sigmoid function eq. (18)
            else if $(t \leq \frac{3T}{4}$ and $t > \frac{T}{2})$ **then**
                Use equation (7) for Hunting coordination
                Update position using the sigmoid function eq. (18)
            else
                Use equation (7) for Hunting cooperation
                Update position using the sigmoid function eq. (18)
            end if
        end for
    end for
    $t = t + 1$
end while
Return the best solution

We will utilize evaluation metrics like ROUGE1, ROUGE2, and ROUGE-L to assess the performance. These metrics are commonly used to evaluate the resemblance of n-grams and the longest shared subsequences in the generated summaries compared to the reference texts. The experiments were conducted using particle sizes of 10, 100 to observe how varying the number of particles affects the recall, precision, and F-measure of the summarization results. Recall indicates the percentage of relevant information from the reference summary captured by the generated summary. Precision measures the proportion of pertinent information within the generated summary. The F-measure provides a harmonic mean of recall and precision, offering a single metric that balances both aspects.

A. Dataset

The datasets provided by the Document Understanding Conference (DUC) are widely used in text summarization research. These datasets are designed to summarize both single and multiple documents. Each dataset contains a collection of English-language source documents and corresponding gold-standard summaries, which human annotators carefully generate. In this work, we evaluate the performance of the BRSA method by comparing its generated summaries with the reference summaries from the DUC datasets, mainly DUC2001 and DUC2002. Table 1 outlines the key characteristics of these datasets.

TABLE I
DATASET DESCRIPTIONS FOR DUC2001 AND DUC2002

Dataset attributes	DUC2001	DUC2002
Total Document Sets	58	30
Documents per Set	5	5
Source of data	duc.nist.gov	duc.nist.gov
length of summary	80	80

Before summarization, all documents were subjected to standard text preprocessing steps. These include sentence segmentation, tokenization, conversion to lowercase, and stop-word removal. This preprocessing ensures a consistent textual representation and reduces noise prior to sentence scoring and selection.

B. Metrics

The metrics called ROUGE (Recall-Oriented Understudy for Gisting Evaluation) are utilized to evaluate the quality of algorithm-produced summaries by comparing them to human references [30], [31]. There are different types of ROUGE metrics, namely ROUGE-SU, ROUGE-L, and ROUGE-N, each focusing on various aspects of the comparison such as skip-bigram overlap, longest common subsequence, and n-gram overlap as defined by equation (19). Higher ROUGE scores indicate better performance of the summarization algorithm.

$$\text{ROUGE-N} = \frac{\sum_{S \in \text{summ-ref}} \sum_{N\text{-grams} \in S} \text{Number}_{\text{Match}}(N\text{-gram})}{\sum_{S \in \text{summ-ref}} \sum_{N\text{-grams} \in S} \text{Number}(N\text{-grams})} \tag{21}$$

The $\text{Number}_{\text{Match}}(N\text{-gram})$ function indicates how many N-grams are found in both the system and reference summaries, and $\text{Number}(N\text{-grams})$ indicates how many N-grams are found in the reference overview. ROUGE-N recall, precision, and F-measure are calculated using the formulas below. Using the proposed approach, a candidate summary (Csum) is generated and compared to the reference summary (Rsum).

- Recall (Re): Measures the percentage of relevant units (e.g., words) in the reference summary that are also in the generated summary. Higher recall indicates that more of the reference summary's content is covered.

$$\text{Re} = \frac{|C_{\text{sum}} \cap R_{\text{sum}}|}{|C_{\text{sum}}|} \tag{22}$$

- Precision (Pr): Measures the percentage of relevant units in the generated summary that are also in the reference summary. Higher precision indicates that the generated summary contains less irrelevant content.

$$\text{Pr} = \frac{|C_{\text{sum}} \cap R_{\text{sum}}|}{|R_{\text{sum}}|} \tag{23}$$

- F-measure (Fs): An integral measure of quality combines recall and precision into one measure.

$$\text{Fs} = 2 \times \frac{\text{Pr} \times \text{Re}}{\text{Pr} + \text{Re}} \tag{24}$$

C. Experimental Protocol

The proposed BRSA-ESDS approach follows the standard evaluation protocol of the DUC benchmarks. Since extractive summarization is formulated as an unsupervised optimization problem, no training phase or train/test split is required. For each set of documents, the algorithm directly generates a single extractive summary of the source documents.

The summary length is constrained to a maximum of 80 words, in accordance with the official DUC2001 and DUC2002 evaluation guidelines. This constraint is strictly enforced during the optimization process to ensure fair and consistent comparison with existing methods. The generated summaries are evaluated by comparing them with the corresponding human reference summaries using the ROUGE-1, ROUGE-2, and ROUGE-L metrics.

D. Proposed Approach Parameter Settings

The values of control parameters in optimization algorithms are usually problem-specific and require empirical tuning. A universal approach is generally ineffective, so customized experimentation is essential for optimal performance. We evaluated different combinations of K_{max} and NP to determine the best parameter settings for our task, focusing on the maximum iterations K_{max} and the particle count NP. For each combination, we calculated the corresponding fitness value. We deemed the T and NP values that produced the best fitness value to be the most suitable. We will use these optimal values, determined from Table 2, throughout this study. This table also includes the fixed switch probability value used in the original algorithm.

TABLE II
PARAMETER SETTINGS FOR THE FITNESS FUNCTION AND BRSA ALGORITHM.

BRSA / Fitness	Parameter	Value
BRSA algorithm	K_{max}	50
	NP	20-100
	γ	0.1
	β	0.1
	r1	[1.NP]
	r2	[-1,1]
	r3	[-1,1]
	rnd	[0,1]
Fitness function	α	0.33
	β	0.33
	γ	0.33

E. BRSA-ESDS Convergence Study

A convergence study is essential for assessing the effectiveness of the Binary Reptile Search Algorithm (BRSA) in extractive single-document summarization tasks. This study evaluates how efficiently BRSA converges toward an optimal solution for ESDS problems as the number of iterations and particles increases. The convergence of the fitness function and the F-measure are critical metrics that must be carefully considered.

1) Impact of Population Size on Fitness Function Convergence: Metaheuristic algorithms aim to find optimal solutions through multiple iterations, making the algorithm's convergence crucial. In this section, we discuss the convergence speed of BRSA based on fitness values. To study its convergence speed, we analyzed this method using two datasets, DUC2001 and DUC2002. Figs. 3 and 4 display the convergence curves, indicating that increasing the number of iterations improves BRSA's ability to explore the search space and quickly identify suitable solutions. These figures present the best fitness value of BRSA and the required number of iterations to achieve the optimal solution across different datasets. The BRSA algorithm effectively converged to the best fitness within iterations for the DUC2001 and DUC2002 datasets.

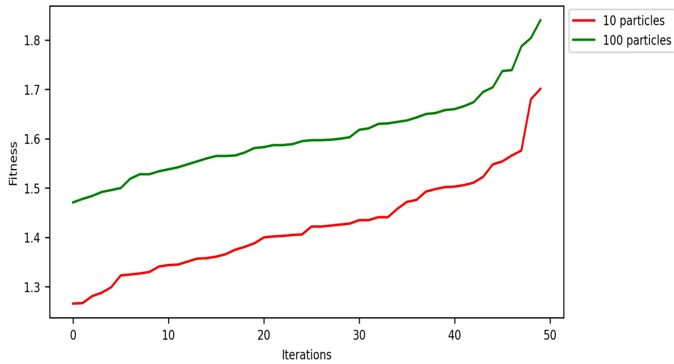


Fig. 3. Variation in fitness across iterations and numbers of particles on the DUC2001 corpus

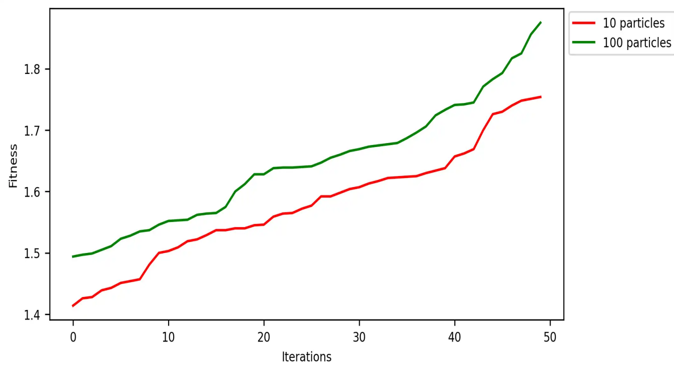


Fig. 4. Variation in fitness across iterations and numbers of particles on the DUC2002 corpus

2) Impact of Population Size and Iterations on F-measure: Iterations and particle counts are critical hyperparameters in swarm-based algorithms. Although performance often improves with increased iterations, a saturation point is typically reached. BRSA, like other swarm algorithms, is sensitive to these parameters. To evaluate BRSA's performance, we varied iterations and population sizes (10 and 100 particles) while measuring the F-measure using Equation (22). Figs. 5-10 illustrate F-measure across corpora and ROUGE metrics. As shown, the proposed algorithm consistently improves with increasing iterations until reaching a peak F-measure value.

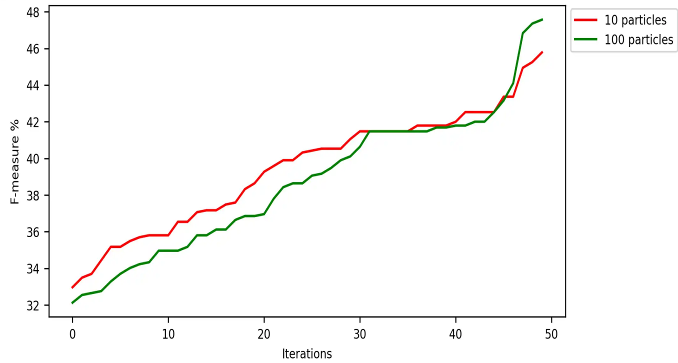


Fig. 5. F-measure variation across iterations and numbers of particles using ROUGE-1 on the DUC2001 corpus.

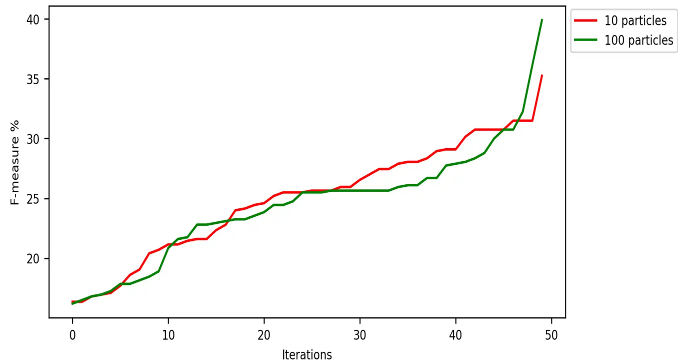


Fig. 6. F-measure variation across iterations and numbers of particles using ROUGE-2 on the DUC2001 corpus.

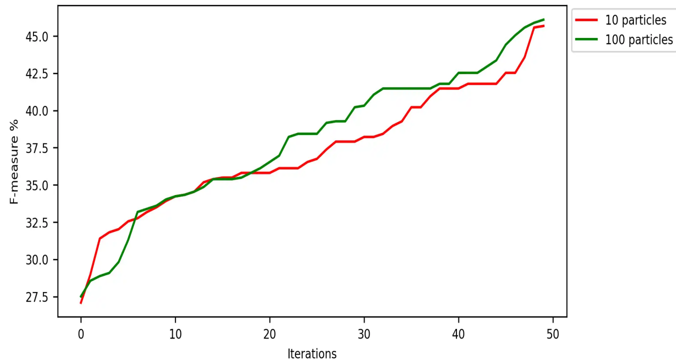


Fig. 7. F-measure variation across iterations and numbers of particles using ROUGE-L on the DUC2001 corpus.

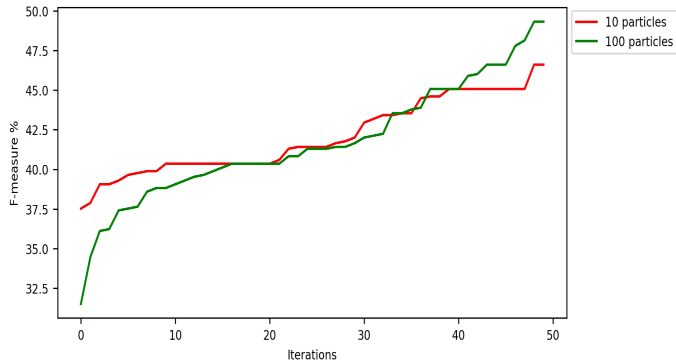


Fig. 8. F-measure variation across iterations and numbers of particles using ROUGE-1 on the DUC2002 corpus.

In our experiments, we used the BRSA to address the ESDS problem. We found that employing a hundred particles produced better results than using ten particles. This indicates that a larger population size within the BRSA framework positively impacted the accuracy of the ESDS system. There are two possible reasons for this observation.

Enhanced Exploration: A larger population (100 particles) allows the BRSA to explore a broader search space within the feature set. This comprehensive exploration may be crucial for identifying the best sentences in the text summary, especially in a large corpus.

Improved Convergence: With more particles, the BRSA has a greater chance of reaching the optimal solution for ESDS. The larger population size enhances the robustness of the search process, potentially preventing the swarm from getting stuck in local optima and guiding it towards the best results. Using 100 particles provided an optimal balance between exploration and convergence for the summarization task.

F. Results Analysis and Discussion

In this section, we will discuss the Binary Reptile Search Algorithm (BRSA) performance on the DUC 2001 and DUC 2002 benchmarks. The performance is evaluated using ROUGE metrics: ROUGE1, ROUGE2, and ROUGE-L. The metrics are displayed for different trials or sets. Each set includes results for recall (Rc), precision (Pc), and F1-score (Fm) (see Tables 3 and 4).

ROUGE1 Results: ROUGE1 evaluates the model's capability to capture important concepts by comparing the unigram similarity in the generated summary with the reference summary. The recall (r) values range from 0.1463 to 0.4615, showing variability in capturing relevant information across different trials. Precision (p) ranges from 0.1690 to 0.4045, indicating a relatively consistent ability to avoid irrelevant information. The F1-score (f) reaches a peak value of 0.4311, suggesting the highest overall quality of summarization in one of the trials. Lower scores, such as 0.1568, indicate underperformance in capturing the essence of the text in some cases.

ROUGE-2 Results: ROUGE-2 measures bigram overlap, providing insights into how well the model captures pairs of consecutive words, a more robust indicator of coherence. The recall ranges from 0 to 0.23, with many values close to 0, implying that the model struggles to capture bigram-level details in several instances. Precision follows a similar trend, ranging from 0 to 0.1933, with a peak F1-score of 0.2100. The relatively low ROUGE2 scores indicate that the algorithm can capture key concepts (as indicated by higher ROUGE1) but struggles with capturing more nuanced or syntactically connected information (bigrams).

ROUGE-L Results: ROUGE-L evaluates the longest matching sequence between the reference and the summary, emphasizing sentence structure. Recall ranges from 0.0976 to 0.4231, with precision from 0.1127 to 0.3708. F1-scores range

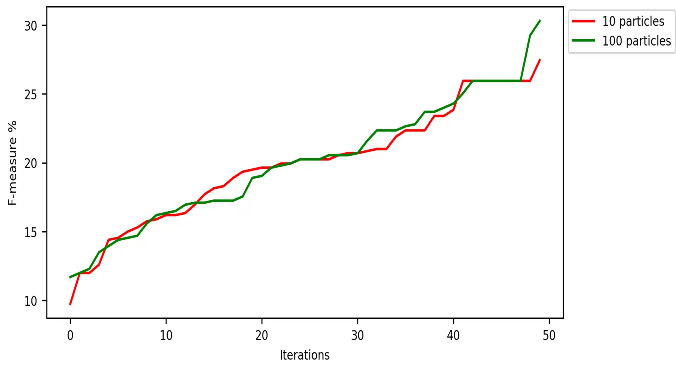


Fig. 9. F-measure variation across iterations and numbers of particles using ROUGE-2 on the DUC2002 corpus.

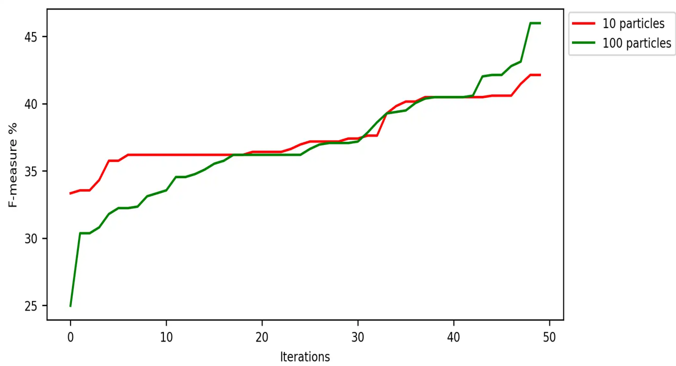


Fig. 10. F-measure variation across iterations and numbers of particles using ROUGE-L on the DUC2002 corpus.

from 0.1046 to 0.3952. Like ROUGE-1, the highest recall and F1-scores in ROUGE-L show that the algorithm is relatively effective at preserving sentence structure in some instances but performs poorly in others (e.g., F1-scores of 0.1046 indicate difficulty in ensuring linguistic coherence).

General Observations: The highest ROUGE1, ROUGE2, and ROUGE-L scores consistently come from one trial (where ROUGE-1 f = 0.4311, ROUGE-2 f = 0.2100, ROUGE-L f = 0.3952). This suggests that the algorithm performs optimally in some instances but inconsistently across trials. Low scores (particularly in ROUGE-2) highlight potential limitations in capturing more complex relationships between words, which could suggest areas for algorithm improvement. Overall, while BRSA performs well in capturing key terms (ROUGE-1), its performance in capturing more complex sentence-level and bigram relations (ROUGE-2 and ROUGE-L) can be inconsistent, pointing to opportunities for refinement. The variability across trials indicates that further fine-tuning or adaptation may be required to achieve consistent performance across all dimensions.

TABLE III
BRSA PERFORMANCE ON THE DUC 2001 ACROSS ITERATIONS.

Iterations	ROUGE Type	Recall	Precision	F-measure
1	ROUGE-1	0.24359	0.22619	0.234568
1	ROUGE-2	0.059406	0.053571	0.056338
1	ROUGE-L	0.217949	0.202381	0.209877
2	ROUGE-1	0.277778	0.308642	0.292398
2	ROUGE-2	0.07767	0.078431	0.078049
2	ROUGE-L	0.255556	0.283951	0.269006
3	ROUGE-1	**0.461538**	0.404494	**0.431138**
3	ROUGE-2	**0.23**	**0.193277**	**0.210046**
3	ROUGE-L	**0.423077**	**0.370787**	**0.39521**
4	ROUGE-1	0.268293	0.244444	0.255814
4	ROUGE-2	0.067961	0.058824	0.063063
4	ROUGE-L	0.231707	0.211111	0.22093
5	ROUGE-1	0.247059	0.247059	0.247059
5	ROUGE-2	0.051546	0.044643	0.047847
5	ROUGE-L	0.188235	0.188235	0.188235
6	ROUGE-1	0.146341	0	0.097561
6	ROUGE-2	0.169014	0	0.112676
6	ROUGE-L	0.156863	0	0.104575
7	ROUGE-1	0.264368	0.252747	0.258427
7	ROUGE-2	0.05	0.04065	0.044843
7	ROUGE-L	0.229885	0.21978	0.224719

TABLE IV
BRSA PERFORMANCE ON THE DUC 2002 ACROSS ITERATIONS.

Iterations	ROUGE Type	Recall	Precision	F-measure
1	ROUGE-1	0.302632	0.270588	0.285714
1	ROUGE-2	0.075472	0.070667	0.073169
1	ROUGE-L	0.302632	0.270588	0.285714
2	ROUGE-1	0.25	0.273973	0.261438
2	ROUGE-2	0.010526	0.010204	0.010363
2	ROUGE-L	0.1875	0.205479	0.196078
3	ROUGE-1	0.21519	0.239437	0.226667
3	ROUGE-2	0.042553	0.040816	0.041667
3	ROUGE-L	0.189873	0.211268	0.2
4	ROUGE-1	0.233766	0.191489	0.210526
4	ROUGE-2	0.041237	0.031008	0.035398
4	ROUGE-L	0.207792	0.170213	0.187134
5	ROUGE-1	0.402597	0.418919	**0.410596**
5	ROUGE-2	0.210526	0.208333	**0.209424**
5	ROUGE-L	0.38961	0.405405	**0.397351**
6	ROUGE-1	0.275	0.30137	0.287582
6	ROUGE-2	0.117647	0.123711	0.120603
6	ROUGE-L	0.25	0.273973	0.261438
7	ROUGE-1	0.304878	**0.352113**	0.326797
7	ROUGE-2	0.093458	0.106383	0.099502
7	ROUGE-L	0.268293	0.309859	0.287582
8	ROUGE-1	0.304878	0.352113	0.326797
8	ROUGE-2	0.093458	0.106383	0.099502
8	ROUGE-L	0.268293	0.309859	0.287582

G. Comparison with State-of-the-art Approaches

The proposed BRSA-ESDS approach is compared with several well-established extractive summarization methods reported in the literature. For all baseline approaches, we relied on the standard implementations and parameter settings recommended in their original publications. When explicit parameter values were not specified, default settings were adopted. No additional parameter tuning was performed on the DUC datasets to ensure a fair and reproducible comparison. To evaluate our approach, we compared it with several established extractive summarization methods on the DUC 2001 and DUC 2002 datasets in this part. These methods include SumBasic [32], a simple yet effective frequency-based approach with an F-measure of around 0.44; LexRank [33], a graph-based method using eigenvector centrality, which achieves an F-measure of approximately 0.46; and TextRank [34], another graph-based technique inspired by PageRank, with an F-measure of around 0.47. Additionally, Luhn's Algorithm, one of the earlier methods focusing on sentence significance, has an F-measure of around 0.41. In contrast, MEAD [35], a centroid-based method that scores sentences based on multiple features, achieves an F-measure of around 0.43 (see Figs. 11-

12).

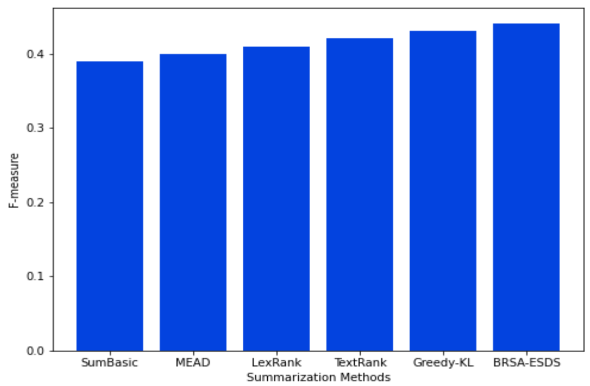


Fig. 11. Comparison of F-measure performance for extractive summarization methods on the DUC2001 corpus.

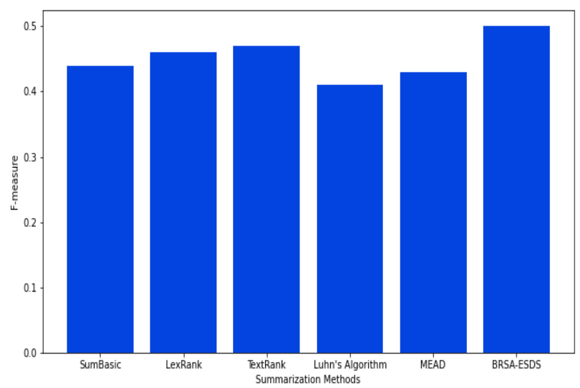


Fig. 12. Comparison of F-measure performance for extractive summarization methods on the DUC2002 corpus.

H. Statistical Significance Analysis

To assess the reliability and robustness of the obtained improvements, statistical significance testing was conducted using a non-parametric bootstrap resampling approach, which is widely adopted in automatic text summarization evaluation. For each ROUGE metric (ROUGE-1, ROUGE-2, and ROUGE-L), 10,000 bootstrap samples were generated from the original document-level scores, and 95% confidence intervals were computed for the mean performance.

On the DUC2001 dataset, the proposed BRSA-ESDS method achieved mean ROUGE-1, ROUGE-2, and ROUGE-L scores of 0.2334, 0.0637, and 0.2105, with corresponding 95% confidence intervals of [0.2046, 0.2632], [0.0494, 0.0784], and [0.1835, 0.2367], respectively.

On the DUC2002 dataset, the obtained mean ROUGE-1, ROUGE-2, and ROUGE-L scores were 0.2550, 0.0588, and 0.2205, with 95% confidence intervals of [0.2318, 0.2795], [0.0462, 0.0721], and [0.1957, 0.2469], respectively.

These confidence intervals indicate that the observed performance gains are statistically stable and not attributable to random variation. Therefore, the statistical analysis supports the effectiveness and robustness of the proposed BRSA-ESDS approach for extractive single-document summarization.

VI. Conclusion and Future Work

In this paper, we introduced a new method for extractive single-document summarization. We model this method as an optimization problem using the Reptile Search Algorithm (RSA) and its binary variant, BRSA-ESDS. Inspired by crocodile hunting behavior, the algorithm effectively balances readability, consistency, diversity, and coverage in generating summaries. By optimizing an objective function that considers these factors, our method improves the quality of generated summaries and controls summary length. We conducted comprehensive evaluations using the ROUGE metric on two benchmark datasets, DUC2001 and DUC2002. Our method demonstrated superior performance compared to several state-of-the-art techniques. These results validate the potential of BRSA-ESDS in addressing the challenges of automated text summarization, providing high-quality, concise summaries that retain vital information from the original text. In future work, we plan to refine the algorithm by incorporating more sophisticated linguistic features and exploring its adaptability to other text summarization tasks, such as multi-document summarization or domain-specific applications. Ultimately, our proposed method lays the groundwork for further advancements in nature-inspired algorithms for text summarization.

References

[1] M. Gambhir and V. Gupta, "Recent automatic text summarization techniques: a survey," *Artificial Intelligence Review*, vol. 47, no. 1, pp. 1–66, 2017, doi: 10.1007/s10462-016-9475-9.
[2] M. A. Fattah, "A hybrid machine learning model for multi-document summarization," *Applied Intelligence*, vol. 40, pp. 592–600, 2014, doi: 10.1007/s10489-013-0490-0.
[3] M. Yousefi-Azar and L. Hamey, "Text summarization using unsupervised deep learning," *Expert Systems with Applications*, vol. 68, pp. 93–105, 2017, doi: 10.1016/j.eswa.2016.10.017.
[4] A. Abuobieda, N. Salim, Y. J. Kumar, and A. H. Osman, "An improved evolutionary algorithm for extractive text summarization," in *Intelligent Information and Database Systems: 5th Asian Conference, ACIIDS 2013*, Kuala Lumpur, Malaysia, Mar. 18-20, 2013, pp. 78–89, doi: 10.1007/978-3-642-36543-0_9.
[5] M. F. Mridha, A. A. Lima, K. Nur, S. C. Das, M. Hasan, and M. M. Kabir, "A survey of automatic text summarization: Progress, process and challenges," *IEEE Access*, vol. 9, pp. 156043–156070, 2021, doi: 10.1109/ACCESS.2021.3129786.
[6] D. V. P. Kumar, S. S. Raj, P. Verma, and S. Pal, "Extractive Text Summarization using Meta-heuristic Approach," in *FIRE (Working Notes)*, pp. 464–474, 2022.
[7] L. Abualigah, M. Abd Elaziz, P. Sumari, Z. W. Geem, and A. H. Gandomi, "Reptile Search Algorithm (RSA): A nature-inspired meta-heuristic optimizer," *Expert Systems with Applications*, vol. 191, p. 116158, 2022, doi: 10.1016/j.eswa.2021.116158.
[8] D. Marcu, "Discourse-based summarization in duc-2001," in *Proceedings of the Document Understanding Conference (DUC01)*, 2001.
[9] P. Over and W. Liggett, "Introduction to duc-2002: an intrinsic evaluation of generic news text," in *Document Understanding Conference*, 2002.
[10] S. Lamsiyah, A. El Mahdaouy, S. O. El Alaoui, and B. Espinasse, "A supervised method for extractive single document summarization based on sentence embeddings and neural networks," in *Advanced Intelligent Systems for Sustainable Development (AI2SD'2019) Volume 4*, pp. 75–88, 2020, doi: 10.1007/978-3-030-36674-2_8.

[11] V. Parikh, V. Mathur, P. Mehta, N. Mittal, and P. Majumder, ”Lawsum: A weakly supervised approach for indian legal document summarization,” *arXiv preprint arXiv:2110.01188*, 2021, doi: 10.1007/978-981-13-8934-4_4.

[12] F. S. Bao, H. Li, G. Luo, M. Qiu, Y. Yang, Y. He, and C. Cen, ”SueNes: A Weakly Supervised Approach to Evaluating Single-Document Summarization via Negative Sampling,” *arXiv preprint arXiv:2005.06377*, 2020, doi: 10.48448/84vs-vg80.

[13] M. Afsharizadeh and S. J. Mousavirad, “A Survey on Transformer-Based Extractive Summarization Methods,” in *Proc. 2024 19th Iranian Conference on Intelligent Systems (ICIS)*, pp. 263–271, 2024, IEEE. doi: 10.1109/ICIS64839.2024.10887480.

[14] J. Dan, W. Hu and Y. Wang, “Enhancing legal judgment summarization with integrated semantic and structural information,” *Artificial Intelligence and Law*, vol. 33, no. 2, pp. 271–292, 2025. doi: 10.1007/s10506-023-09381-8.

[15] R. Srivastava, P. Singh, K. P. S. Rana, and V. Kumar, ”A topic modeled unsupervised approach to single document extractive text summarization,” *Knowledge-Based Systems*, vol. 246, p. 108636, 2022, doi: 10.1016/j.knosys.2022.108636.

[16] M. Isonuma, J. Mori, and I. Sakata, ”Unsupervised neural single-document summarization of reviews via learning latent discourse structure and its ranking,” *arXiv preprint arXiv:1906.05691*, 2019, doi: 10.18653/v1/P19-1206.

[17] W. Song, L. C. Choi, S. C. Park, and X. F. Ding, ”Fuzzy evolutionary optimization modeling and its applications to unsupervised categorization and extractive summarization,” *Expert Systems with Applications*, vol. 38, no. 8, pp. 9112–9121, 2011, doi: 10.1016/j.eswa.2010.12.102.

[18] M. Azam, S. Khalid, S. Almutairi, H. A. Khattak, A. Namoun, A. Ali and H. S. M. Bilal, “Current trends and advances in extractive text summarization: A comprehensive review,” *IEEE Access*,2025. doi: 10.1109/TCSS.2025.3583893.

[19] P. Verma, S. Pal, and H. Om, ”A comparative analysis on Hindi and English extractive text summarization,” *ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP)*, vol. 18, no. 3, pp. 1–39, 2019, doi: 10.1145/3308754.

[20] D. Debnath, R. Das, and P. Pakray, ”Extractive single document summarization using multi-objective modified cat swarm optimization approach: ESDS-MCSO,” *Neural Computing and Applications*, pp. 1–16, 2021, doi: 10.1007/s00521-021-06337-4.

[21] S. H. Mirshojaee, B. Masoumi, and E. Zeinali, ”MAMHOA: a multi-agent meta-heuristic optimization algorithm with an approach for document summarization issues,” *Journal of Ambient Intelligence and Humanized Computing*, vol. 11, no. 11, pp. 4967–4982, 2020, doi: 10.1007/s12652-020-01776-8.

[22] P. Verma and A. Verma, ”A review on text summarization techniques,” *Journal of Scientific Research*, vol. 64, no. 1, pp. 251–257, 2020, doi: 10.37398/JSR.2020.640148.

[23] R. Rautray and R. C. Balabantaray, ”Cat swarm optimization based evolutionary framework for multi document summarization,” *Physica A: Statistical Mechanics and Its Applications*, vol. 477, pp. 174–186, 2017, doi: 10.1016/j.physa.2017.02.056.

[24] R. S. Selvan and K. Arutchelvan, ”An efficient social spider optimization algorithm based multi-document summarization model,” in *2021 6th International Conference on Inventive Computation Technologies (ICICT)*, pp. 855–860, IEEE, 2021, doi: 10.1109/ICICT50816.2021.9358521.

[25] M. Umair, A. Khan, F. Ullah, A. Masmoudi and M. Faheem, “Global and Local Context Fusion in Heterogeneous Graph Neural Network for Summarizing Lengthy Scientific Documents,” *IEEE Access*,2025. doi: 10.1109/ACCESS.2025.3553755.

[26] H. Siranjeevi, S. Venkatraman and S. P. Raja, “EBPGA: Extractive Text Summarization Using Binary Particle Swarm Optimization and Masked Genetic Algorithm,” *IEEE Transactions on Computational Social Systems*, 2025. doi: 10.1109/TCSS.2025.3583893.

[27] R. A. Khan, B. Sabir, A. Sarwar, H. Liu, and C. H. Lin, ”Reptile search algorithm (RSA)-based selective harmonic elimination technique in packed E-Cell (PEC-9) inverter,” *Processes*, vol. 10, no. 8, p. 1615, 2022, doi: 10.3390/pr10081615.

[28] D. Wu, C. Wen, H. Rao, H. Jia, Q. Liu, and L. Abualigah, ”Modified reptile search algorithm with multi-hunting coordination strategy for global optimization problems,” *Math. Biosci. Eng*, vol. 20, no. 6, pp. 10090–10134, 2023, doi: 10.3934/mbe.2023443.

[29] B. Sasmal, A. G. Hussien, A. Das, K. G. Dhal, and R. Saha, ”Reptile search algorithm: Theory, variants, applications, and performance evaluation,” *Archives of Computational Methods in Engineering*, vol. 31, no. 1, pp. 521–549, 2024, doi: 10.1007/s11831-023-09990-1.

[30] C. Y. Lin, ”Rouge: A package for automatic evaluation of summaries,” in *Text Summarization Branches Out*, pp. 74–81, 2004.

[31] K. Ganesan, ”Rouge 2.0: Updated and improved measures for evaluation of summarization tasks,” *arXiv preprint arXiv:1803.01937*, 2018, doi: 10.18653/v1/2022.findings-acl.122.

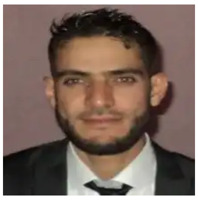
[32] L. Vanderwende, H. Suzuki, C. Brockett, and A. Nenkova, ”Beyond SumBasic: Task-focused summarization with sentence simplification and lexical expansion,” *Information Processing & Management*, vol. 43, no. 6, pp. 1606–1618, 2007, doi: 10.1016/j.ipm.2007.01.023.

[33] G. Erkan and D. R. Radev, ”Lexrank: Graph-based lexical centrality as salience in text summarization,” *Journal of Artificial Intelligence Research*, vol. 22, pp. 457–479, 2004, doi: 10.1613/jair.1523.

[34] R. Mihalcea and P. Tarau, ”Textrank: Bringing order into text,” in *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pp. 404–411, 2004.

[35] D. R. Radev, S. Blair-Goldensohn, and Z. Zhang, ”Experiments in single and multidocument summarization using MEAD,” in *First Document Understanding Conference*, pp. 1–7, 2001.

Abdelaali Bekhouche is a lecturer in the Department of Computer Science at Abbes Laghrour University, Khenchela, Algeria. He received his Ph.D. in Computer Science from Badji Mokhtar University, Annaba, and obtained his Habilitation to Direct Research (HDR) in 2022. His research interests include Natural Language Processing, Word Sense Disambiguation, and swarm intelligence–based optimization.



Mohamed Boussalem received his Ph.D. in Computer Science from the University of Batna, Algeria, in 2024. He is currently an Associate Professor at Abbas Laghrour University, Khenchela, Algeria. His research focuses on natural language processing, machine learning, text summarization, misinformation, classification, and swarm-based optimization algorithms.

Hichem Haouassi received his Ph.D. degree in Computer Science from the University of Batna, Algeria, in 2012. He is currently a Full Professor at Abbas Laghrour University, Khenchela, where he contributes to teaching and research activities in Artificial I ntelligence a nd d ata-driven decision systems. His main research interests include Data Mining, Metaheuristics, swarm-based optimization techniques, and Feature Selection.