



# MAMHOA: a multi-agent meta-heuristic optimization algorithm with an approach for document summarization issues

Seyed Hossein Mirshojaee<sup>1</sup> · Behrooz Masoumi<sup>1</sup> · Esmail Zeinali<sup>1</sup>

Received: 25 April 2019 / Accepted: 6 February 2020 / Published online: 14 March 2020  
© Springer-Verlag GmbH Germany, part of Springer Nature 2020

## Abstract

Today, given the increasing volume of information and the difficulty of using them for specific applications such as email, websites, news, etc., the use of automated information summarization algorithms has become more popular than traditional algorithms. Taking advantage of computer algorithms, these algorithms produce a summary of information while retaining its original meaning. However, given its semantic and structural properties as well as the variable comparative parameters of information, the summarization process is considered as an NP-hard problem. Therefore, to solve these problems it is better to use meta-heuristic algorithms, which are generally inspired by the behavior of nature. These meta-heuristic algorithms help to better solve the hard problems through producing optimum solutions. In this paper, we propose an optimization algorithm named multi-agent meta-heuristic optimization algorithm (MAMHOA) for extractive text summarization. MAMHOA is a combination of biogeography-based optimization (BBO) algorithm and multi-agent systems concepts to generate an optimum summary. Several computational tests are used to evaluate the effectiveness and efficiency of the proposed algorithm, which is compared to other algorithms provided in the literature. MAMHOA and other algorithms are tested on DUC2002 datasets and attained solutions are analyzed using ROUGE metrics. From the results obtained, it can be seen that the proposed algorithm is more effective and efficient than those mentioned in the literature i.e. Baseline and state-of-the-art methods for different ROUGE metrics.

**Keywords** NP-hard problems · Meta-heuristic algorithms · Multi-agent meta-heuristic optimization algorithm · Biogeography-based optimization · Extractive text summarization

## 1 Introduction

The goal of optimization is to get the best or near the best acceptable solution, given the constraints and needs of the problem. For a problem, there may be several different solutions that are defined to be compared and then an optimal solution is selected. Research has shown that meta-heuristic algorithms and multi-agent systems can be used as a solution for improving the optimization process. Meta-heuristic

algorithms are algorithms that are inspired by nature, physics, and humans and are used to solve many optimization problems. Optimization algorithms are usually used in combination with other algorithms to reach the optimal solution or to exit from the local optimal solution. Multi-agent systems are among the new approaches that have been proposed to achieve the above-mentioned capabilities. Agent-based systems in the field of distributed artificial intelligence have been studied for several years and have been used in various fields of computer science. Agents are able to perceive and react against the environment, having an extensive range of meanings depending on the problem to be solved. Generally, an agent can be characterized by the following four properties: (1) living and acting in an environment; (2) having the ability to sense its local surroundings; (3) acting based on a specific purpose; and (4) having some reactive behaviors (Li and Jing 2016). In a summarization problem, the goal is to create the best or near the best optimum summary of one document or more documents, which consist of three

✉ Behrooz Masoumi  
Masoumi@qiau.ac.ir

Seyed Hossein Mirshojaee  
Mirshojaee.h@gmail.com

Esmail Zeinali  
Zeinali@qiau.ac.ir

<sup>1</sup> Faculty of Computer and Information Technology  
Engineering, Qazvin Branch, Islamic Azad University,  
Qazvin, Iran

features obtained by extracting the most important contents. These features are as follows (Oufaida et al. 2014):

- *Coverage* All the information of important topics in the documents should be covered by the summary.
- *Non-redundancy* It indicates the novelty in the summary.
- *Relevancy* All the information of contents that are relevant to a user should be contained in the summary.

In general, summarization algorithms can be categorized into two classes, namely, extractive and abstractive summarization (Ferreira et al. 2013). Extractive summarization has to do with concatenating multiple relevant sentences which are chosen without any modification, i.e. exactly as they are used in the original document. Moreover, creating abstractive summaries is a difficult task since it requires paraphrasing sections of the source document. Therefore, it needs natural language generation tools. Furthermore, abstractive algorithms may reuse clauses or phrases from original documents (Nenkova et al. 2011). In this paper, we proposed an algorithm named MAMHOA algorithm, which is a combination of a biogeography meta-heuristic algorithm and multi-agent systems aimed at improving extractive text summarization. In MAMHOA, a set of agents i.e. habitats, is considered which are competing or cooperating with their neighbors to find the best conditions and best habitats. The agents collaborate in a multi-agent system to achieve a specific, shared goal, and each agent performs its own task. This collaboration can be done through communication and exchange of information with neighbors of each agent. The performance of the proposed algorithm is evaluated for single-document extractive text summarization on DUC 2002<sup>1</sup> datasets. As for evaluation, the proposed algorithm is compared with some of the Baseline and state-of-the-art methods of extractive text summarization.

The rest of the paper is organized as follows: In Sect. 2, the literature review is discussed. In Sect. 3, the original Biogeography-based optimization is explained. In Sect. 4, the proposed algorithm, i.e. MAMHOA is introduced and assessed on the single-document extractive text summarization. In Sect. 5, the results of experiments are presented and discussed, and finally in Sect. 6 the paper is concluded.

## 2 Literature review

Our work generates a new meta-heuristic optimization algorithm for single-document extractive text summarization based on multi-agent concepts. In the literature, we will briefly review some of the methods in field of text

summarization. As a first algorithm, Luhn (1958) is used for text summarization based on statistical features such as the frequency of words and the relative positioning of each sentence in the document. The authors in (Mihalcea and Tarau 2004) introduced a graph-based technique called TextRank. This algorithm represents the document in the form of a graph of sentences, and the edges between two sentences are connected based on the similarity between them. LexRank (Radev and Erkan 2004) applied the notion of eigenvector centrality to the graph representation of sentences, with the aim of estimating the importance of each sentence in the document. Moreover, Steinberger and Jezek (2004) put forth a topic-based algorithm named Latent Semantic Analysis (LSA), which seeks to spot salient sentences in the document through applying Singular Value Decomposition (SVD) over document matrix  $D$  of size  $m \times n$ , consisting of  $m$  sentences and  $n$  number of terms. Nenkova and Vanderwende (2005) developed a greedy search approximation algorithm called SumBasic that makes use of a frequency-based sentence selection component with a component to re-weight the word probabilities in order to minimize redundancy. Amancio and Tohalino (2018) proposed an adapted model, which includes sentences and edges, which are created given the number of shared words between sentences. The proposed algorithm distinguishes between edges linking sentences in different documents (interlayer) and those connecting sentences from the same document (intra-layer). Rautray and Balabantaray (2018) put forth a new Cuckoo search-based multi-document summarizer (MDSCSA) to deal with the problem of multi-document summarization. They compare the new MDSCSA with two other nature-based summarization techniques, including Particle swarm optimization based summarization (PSOS) and Cat swarm optimization based summarization (CSOS). Haghighi and Vanderwende (2009) developed an algorithm for text summarization, which generates the output summary of the document because of the criteria of Kullback-Lieber (KL) divergence. As a phrase-based summarization technique, Integer Linear Programming (Woodsend and Lapata 2010) encodes the grammaticality constraints across phrase dependencies using integer linear programming. Wan (2010) proposed URANK, which is a unified rank methodology based on graph model. It is able to handle single-document and multi-document summarization simultaneously. Asgari and Masoumi (2013) provided an algorithm to improve the performance of text summarization using bacterial foraging optimization algorithm. Egraph (Parveen et al. 2015) is considered as an entity graph-based system, which makes use of a bipartite graph of sentence and entity nodes for document representation. They developed a system called “Egraph + Coh” which includes a coherence measure calculated by using the averaged-out degree of an unweighted projection graph. Tgraph (Parveen et al. 2015) is an unsupervised graph-based text summarization algorithm

<sup>1</sup> <http://www-nlpir.nist.gov/projects/duc/guidelines/2002.html>.

that applies Latent Dirichlet Allocation (LDA) for topic modeling. “Tgraph + Coh” takes account of coherence calculated using weighted projection graph for generating a summary. Cheng and Lapata (2016) developed NN-SE, which is a neural summarization model for sentence extraction. SummaRuNNer (Nallapati et al. 2016) is a recent extractive text summarization algorithm, which is based on RNNs. Mirshojaee et al. (2017) used an original biogeography-based optimization algorithm for extractive text summarization. The proposed algorithm is inspired by the natural migration of species between habitats.

CoRank (Fang et al. 2017) is a graph-based unsupervised extractive text summarization approach, which involves combining word-sentence relationship with the graph-based ranking model. Now we discuss some approaches, which use BBO meta-heuristic algorithm in field of optimization. Zhang et al. (2018) presented a novel hybrid algorithm based on BBO and Grey wolf optimizer (GWO), named as HBBOG. In the proposed algorithm, both algorithms (BBO and GWO) are improved. Yang and Li (2017) introduced an improved biogeography-based optimization algorithm that uses the nonlinear migration operator. They have used this proposed algorithm in mobile robot’s path planning. Chen and Tianfield (2016) developed a covariance matrix-based migration (CMM) to reduce BBO’s dependence on the coordinate system, resulting in the enhancement of BBO’s rotational invariance. Khishe et al. (2017) proposed a novel exponential-logarithmic migration operator for the original BBO, named as ELBBO. Paraskevopoulos et al. (2017) proposed a modified BBO named as real-coded biogeography-based optimization (RCBBO) that combined fuzzy decision making for the cognitive radio engine design used in the internet of things (IoT). To improve the population diversity, the authors employed Gaussian, Cauchy and Levy mutations as the mutation operator. Feng et al. (2017) proposed a modified BBO named as PRBBO that was improved for solving global optimization problems. In the modified algorithm, triple combination includes, migration operator combined with random ring topology, a modified mutation operator, and a self-adaptive Powell’s algorithm. In the proposed algorithm, local ring topology is used instead of the global topology to increase population diversity. Lohokare et al. (2013) presented a mimetic BBO named as ABBOMDE, in which the performance of BBO is accelerated with the help of a modified mutation and clear duplicate operators.

### 3 Biogeography-based optimization algorithm

In this section, we introduce the original BBO algorithm, which is used in the proposed algorithm. BBO is a new intelligence-based meta-heuristic algorithm inspired by

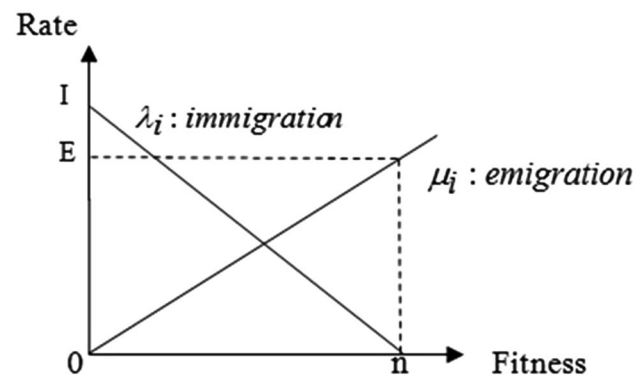


Fig. 1 The curve of BBO Linear Migration (Simon 2008)

nature based on the concept of animal migration to find the best habitat (Simon 2008; Harish 2015). In BBO, each habitat represents a candidate solution of the problem. Species migrate from one habitat to another habitat depending on a suitability index called habitat suitability index (HSI). HSI actually represents the individual’s fitness. Parameters such as rainfall, temperature, region, humidity, etc., affect the excellent characteristics of biological habitats. In BBO algorithm, from Simon’s point of view, these characteristics are called suitability index variables (SIVs). Simply  $n$ -dimensional habitat, which is a candidate solution of the problem, is formed by  $n$  SIVs whose fitness is denoted by HSI. In BBO (Fig. 1), each habitat has an immigration rate ( $\lambda$ ) which means that it has a desire to accept poor habitats, and emigration rate ( $\mu$ ) which means that there is a strong tendency to migrate to rich habitat. In BBO, the solution is improved with a focus on the immigration and emigration of solution features within habitats. High HSI habitats share their good features with low HSI habitats, with low HSI habitats accepting the novel features of high HSI habitats. In accordance with the principle of the BBO, it is highly probable that good solutions share their SIVs with other solutions and poor solutions accept SIVs from other solutions. There is a decrease in emigration rates from high HSI habitats to low HSI ones so that the highest HSI habitat has the maximum emigration rate while there is an increase in immigration rate from high HSI habitats to low HSI ones, with the highest HSI habitat having the lowest immigration rate.

Immigration rate  $\lambda$  and emigration rate  $\mu$  are calculated according to the formulas (1) and (2).

$$\lambda_i = I \left( 1 - \frac{k_i}{n} \right) \quad (1)$$

$$\mu_i = E \left( \frac{k_i}{n} \right) \quad (2)$$

The parameters of  $\lambda_i$  and  $\mu_i$  are the immigration rate and emigration rate for  $i$ th habitat, respectively. The parameters of  $I$  and  $E$  are the maximum immigration rate and maximum emigration rate, respectively, and  $n$  denotes the population size. The  $k_i$  stands for the fitness rank of  $i$ th habitat after fitness of  $i$ th habitat is sorted, so that worst solution has  $k_i$  of 1 and the best solution has  $k_i$  of  $n$ . Migration and mutation are two crucial operators in BBO. The migration operator is responsible for generating new solution at each iteration and is similar to the crossover operator in the evolutionary algorithm. The mutation operator is done randomly on habitats and is responsible for maintaining the diversity of habitats and preventing the trapping of the algorithm in the local optimal point. According to formulas (3) and (4), the parameter of  $P_s$  is the probability which denotes  $s$  species in the habitat ranging from  $t$  to  $(t + \Delta t)$  where  $\lambda_s$  is immigration rate and  $\mu_s$  is emigration rate when there are  $s$  species in the habitat.

$$P_s(t + \Delta t) = P_s(t)(1 - \lambda_s \Delta t - \mu_s \Delta t) + P_{s-1} \lambda_s \Delta t + P_{s+1} \mu_{s+1} \Delta t \tag{3}$$

$$p_s = \begin{cases} (\lambda_s + \mu_s)p_s + \mu_{s+1}p_{s+1} \cdots s = 0 \\ (\lambda_s + \mu_s)p_s + \lambda_{s-1}p_{s-1} + \mu_{s+1}p_{s+1} \cdots 1 \leq s \leq S_{\max} - 1 \\ (\lambda_s + \mu_s)p_s + \mu_{s-1}p_{s-1} \cdots S = S_{\max} \end{cases} \tag{4}$$

At time  $t + \Delta t$ , one of the following conditions must hold for  $s$  species in the habitat.

1. If there were  $s$  species in the habitat at time  $t$ , then there is no immigration and no emigration of species within time  $t$  and  $t + \Delta t$ .
2. If there were  $(s - 1)$  species in the habitat at time  $t$ , then there is one species immigrated within time  $t$  and  $t + \Delta t$ .
3. If there were  $(s + 1)$  species in the habitat at time  $t$ , there is one species emigrated within time  $t$  and  $t + \Delta t$ .

To ignore the probability of more than one immigration or emigration, we set  $\Delta t$  very small so that  $\Delta t \rightarrow 0$ .

Solutions with very high HIS and very low HSI are equally improbable. Medium HSI solutions are relatively probable. The mutation rate is expressed by formula (5) where  $m_{\max}$  denotes a user-defined parameter,  $p_i$  denotes priori existence probability, and  $p_{\max} = \max \{p_i, i = 1, 2, \dots, p_s\}$ .

$$m_i = m_{\max} \left( 1 - \frac{p_i}{p_{\max}} \right) \tag{5}$$

In the case of the original mutation operator, a random value, created in the whole solution space, is used to probabilistically substitute SIV in each solution. Mutation represents the sudden changes in the habitat. This operator

```

Randomly initialize a Population of n Habitats  $H_i, i \leftarrow \{1, \dots, n\}$ ;
Initialize Max Iteration;
1. while the Termination Criteria are not Satisfied do
2.   Computing Fitness (HSI) for each Habitat and sort Habitats according their HSI;
3.   for  $i \leftarrow 1$  to  $n$  do // * n is Population size
4.     a. Computing  $\lambda$  and  $\mu$  for each Habitat Based on HSI;
5.   end
Migration:
6.   Select  $H_i$  with Probability based on  $\lambda_i$ ;
7.   if  $H_i$  is selected then
8.     a. select  $H_j$  with Probability based on  $\mu_j$ ;
9.   if  $H_j$  is selected then
10.    a. Randomly Select SIV from  $H_j$ ;
11.    b. Replace SIV in  $H_i$  with one from  $H_j$ ;
12. end
Mutation:
13. Select  $H_i$  with Probability based on the Mutation rate;
14. if  $H_i$ (SIV) is selected then
15.   a. Perform Mutation;
16. end
17. Evaluate the Fitness values of the Habitats;
18. Perform Elitism and Update the Best Solution;
19. end
20. return Best Solution;
    
```

Fig. 2 Pseudo code of BBO algorithm

is tasked with maintaining the diversity in the population during BBO process. Mutation randomly makes minor changes to habitat SIVs based on the habitat's a priori probability. The algorithm in Fig. 2, indicates the pseudo-code of BBO algorithm with mutation and migration operators, where  $H_i$  and  $H_j$  represent the  $i$ th habitat and the  $j$ th habitat, respectively.  $N$  denotes the highest number of species and  $D$  denotes the dimension of a solution.

Finally, following the migration and mutation, the specific habitat is compared against the criterion habitat. If the HSI value of this specific habitat exceeds that of the criterion habitat, the latter is substituted by the former. Otherwise, the criterion habitat should be preserved as a better option. By continuously iterating evolution, the algorithm preserves the better conditions, improving the poor individual and guiding the search process towards the global optimal individual approximation.

### 4 Proposed MAMHOA algorithm

In this section, we present our proposed algorithm, i.e. a multi-agent meta-heuristic optimization algorithm (MAMHOA) with an approach for single-document extractive text summarization. Given that the proposed algorithm uses BBO algorithm, firstly, we need to consider the following three initial important definitions:

1. In MAMHOA, each agent represents a habitat in BBO, which is a candidate solution.

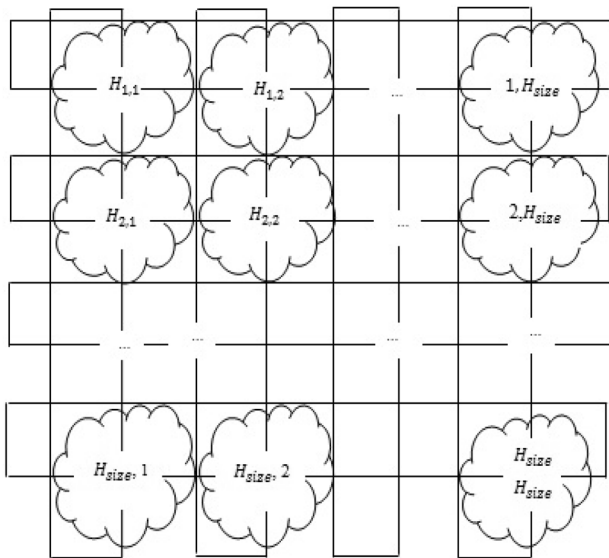


Fig. 3 The habitats lattice

2. In MAMHOA, all habitats live in a lattice-like environment, where each habitat is placed at one point of the lattice. These habitats seek to increase habitat suitability index (HSI) as well as efficiency through competing or cooperating with their neighbors.
3. The lattice dimension is  $H_{size} \times H_{size}$  so that habitats can be placed on this lattice as shown in the Fig. 3.

According to formulas (6) and (7), habitat location is defined as row and column ( $r, c$ ), and represented as  $H_{r,c}$ ,  $r, c = 1, 2, 3 \dots H_{size}$  in lattice, so the neighbors of a habitat are defined as follows:

$$\text{Neighbors}_{r,c} = \{H_{r',c}, H_{r,c'}, H_{r,c''}, H_{r'',c}\} \tag{6}$$

$$\begin{aligned} r' &= \begin{cases} r-1 \dots r \neq 1 \\ H_{size} \dots r = 1 \end{cases} & r'' &= \begin{cases} r+1 \dots r \neq H_{size} \\ 1 \dots r = H_{size} \end{cases} \\ c' &= \begin{cases} c-1 \dots c \neq 1 \\ H_{size} \dots c = 1 \end{cases} & c'' &= \begin{cases} c+1 \dots c \neq H_{size} \\ 1 \dots c = H_{size} \end{cases} \end{aligned} \tag{7}$$

Figure 4 shows each habitat along with its four neighbors, which forms a small local environment that can only sense the environment. As already mentioned, habitats compete or cooperate with their neighbors to achieve the final goal, i.e. the best or near the best result. Given that a habitat only senses its local environment, competition and cooperation can only occur between the habitats and their neighbors. This helps to spread the information to the entire lattice. Given that competition and collaboration are done locally, the information transmission is slow across the habitat lattice.

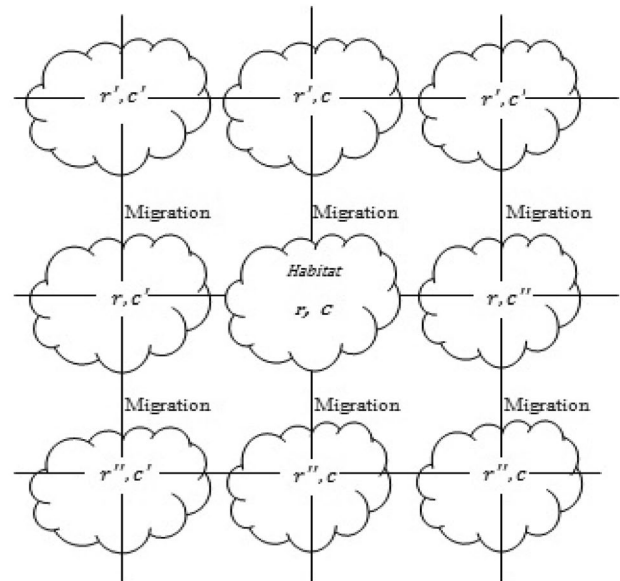


Fig. 4 The habitat neighbors

### 4.1 Habitats selection strategies used by MAMHOA

The following procedure is performed to achieve better solutions more quickly and accurately. Initially, the habitats compete with each other and with their neighbors based on their competencies. Moreover, habitats use the self-learning operator to improve their ability (Figs. 3, 4).

The competition operator is performed on the habitat located at ( $r, c$ ),  $H_{r,c} = (h_1, h_2, \dots, h_n)$ . we consider that the maximum cost function is  $\max_{r,c} = (m_1, m_2, \dots, m_k), k = (1, 2, \dots, n)$  and  $H_{r,c}$  is the habitat neighbor with the maximum habitat suitability index (HSI). Therefore, given the neighbor relations shown in Fig. 4, the best habitat conditions are as follows:  $\max_{r,c} \in \text{Neighbors}_{r,c} \forall \alpha \in \text{Neighbors}_{r,c} f(\alpha) \leq f(\max_{r,c})$ . If the habitat suitability index is represented as  $\text{HSI}(H_{r,c}) \geq \text{HSI}(\max_{r,c})$ , then  $H_{r,c}$  is the winner because its cost function value is greater than the maximum cost function of its neighbors, so it can still live in the habitat lattice and its position does not change in the search space. Otherwise,  $H_{r,c}$  is the loser and the position of  $H_{r,c}$  in the habitat lattice should be occupied by the new habitat ( $\text{New}_{r,c}$ ). The proposed algorithm presents two strategies to select the new habitat.

1. In the first strategy, when  $H_{r,c}$  is the loser, it has still useful information to obtain good solutions. Hence, this strategy as a kind of heuristic crossover involves using the information of loser habitat as well. The new habitat, represented by  $\text{New}_{r,c} = (e_1, e_2, \dots, e_n)$ , is calculated using formula (8) where  $\text{rand}(0, 1)$  is a random number

**Fig. 5** Self-learning operators in MAMHOA algorithm to create best solution

```

Initializing Self-learning operators:
Pc: The crossover probability;
Pm: The mutation probability;
LB: The lower bound variables;
UB: The upper bound variables;
SR: The Radius of Self-Learning;
HSize: The high dimension of habitat lattice;
LSize: The low dimension of habitat lattice;
nVar :The Number of Decision Variables;

Operation:
1. for r←1 to HSize do
2. for c←1 to HSize do
3. if (r=1 & c=1) then
    HbitateLattice(r, c)← AgentLattice(r, c);
    else
4. for k← 1 to nVar do
    a. HbitateLattice(r, c).SIV (k) ←AgentLattice(r, c).SIV (k)...
    b. Generating random numbers (1-SR, 1+SR);
5. end
6. end
7. end

Competition operator for Self-learning:
8. HbitateLattice ← Competition (HbitateLattice, HSize, NeighborSize, nVar, Pc, Pm, LB, UB);
9. for i← 1 to LSize do
10. for c←1 to HSize do
    HbitateLattice(r, c).HSI ← feval (ff,AgentLattice (i, j).SIV, S, sentenceNum, similarityMatrix);
11. if (GlobalAgent.HSI > HbitateLattice (r, c).HSI) then
    a. GlobalAgent.HSI← HbitateLattice (r, c).HSI;
    b. GlobalAgent.SIV ← HbitateLattice (r, c).SIV;
12. end
13. end
14. end

Output: Best habitats
15. BEST(Iter) ← GlobalAgent.HSI;
    
```

in [0, 1] and  $X_{max}, X_{min}$  are the maximum and minimum HSI of habitats, respectively.

$$e_k = \begin{cases} X_{max} \dots \dots \dots e_k \geq X_{max} \\ X_{min} \dots \dots \dots e_k \leq X_{min} \\ m_k + \text{rand}(0, 1).(m_k - h_k), k = 1, 2, \dots, n. \dots \dots \dots \text{Otherwise} \end{cases} \tag{8}$$

2. In second strategy,  $\max_{r,c}$  is first mapped to [0, 1], using formula (9) and new habitat neighbor, i.e.  $\text{New}_{r,c}$  is calculated using formula (10).

$$m_k = \frac{(m_k - X_{\min k})}{(X_{\max k} - X_{\min k})} k = 1, 2, \dots, n \tag{9}$$

$$\text{New}_{r,c} = (m_1, m_2, \dots, m_{i_{1-1}}, m_{i_2}, m_{i_{2-1}}, \dots, m_{i_{1+1}}, m_{i_1}, m_{i_{2+1}}, m_{i_{2+2}}, \dots, m_n) \tag{10}$$

If  $1 < r_1 < n, 1 < r_2 < n$  and  $r_1 < r_2$ . Finally,  $\text{New}_{r,c}$  is obtained through mapping  $\text{New}_{r,c}$  back to  $[X_{min}, X_{max}]$  by formula (11).

$$e_k = X_{\min k} + e'_k.(X_{\max k} - X_{\min k}), k = 1, 2, \dots, n \tag{11}$$

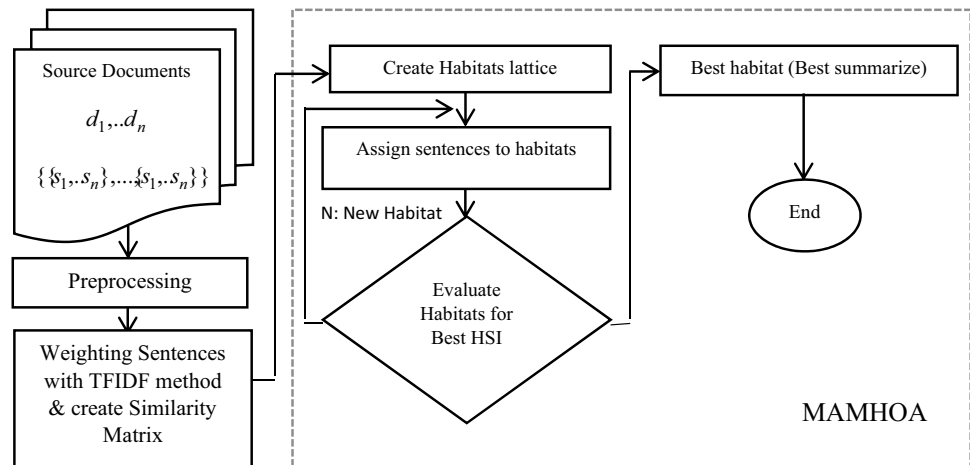
As described above, the first strategy increases exploitation and the second strategy increases the exploration. The  $p_c$  probability parameter determines which of two strategies should be used. That is, if  $\text{rand}(0, 1) < p_c$ , then the first strategy is used, otherwise the second strategy is used. The algorithm in Fig. 5 presents the competition between self-learning operators in MAMHOA algorithm to create best solution. Table 1 shows the initial values of the used parameters in MAMHOA algorithm (Fig. 5). The initial parameters are based on original BBO parameters (Simon 2008).

As shown by the algorithm in Fig. 5 and Table 1, the variables and operators used in the proposed algorithm are as follows: Crossover probability ( $P_c$ ), Mutation probability ( $P_m$ ), Habitat Number (Habitats), Lower limit of decision variables (LB), Upper limit of decision variables (UB), Self-learning Operator (SR). According to algorithm in Fig. 5 and Figs. 3 and 4, initially a  $H_{size} \times H_{size}$  lattice is created. Then, each c and r location is assigned a habitat. Given the SR probability, the random values are generated and assigned to each habitat in terms of optimum conditions. Then, the habitats are compared in terms of their conditions and the habitat with the highest HSI is selected as the best habitat.

**Table 1** Initial parameters used for the MAMHOA algorithm

Parameters	Description	Initial values
nVAR	Decision variables	S = {Sentences number}
VarSize	Matrix size	[1..S]
LB	Lower limit of decision variables	1
UB	Upper limit of decision variables	S
MaxIter	Maximum number of iterations	200
Habitats	Habitats number	100
Lsize, Hsize	Lower and upper limits of habitats lattice	[1..100]
Neighbor size	Number of neighbors for each habitat	4
Pm	Mutation probability	0.2
Pc	Crossover probability	0.3
SR	Self-learning Radius	0.2

**Fig. 6** Documents summarization processing using MAMHOA



Consequently, the migration occurs from a habitat to a new habitat with better conditions.

**4.2 Documents summarization processing**

In this section, the process of documents summarization using MAMHOA algorithm is shown in Fig. 6. According to Fig. 6  $D = \{d_1, d_2, \dots, d_n\}$  are input source documents. Each  $d_i$  consists of a collection of sentences  $S = \{s_1, s_2, \dots, s_n\}$  and each  $s_i$  consists of a collection of words  $W = \{w_1, w_2, \dots, w_n\}$ . The summarization steps are shown as follows:

*Step 1:* Input documents

Reading documents  $D = \{d_1, d_2, \dots, d_n\}$  including n sentences  $S = \{s_1, s_2, \dots, s_n\}$  and n words  $W = \{w_1, w_2, \dots, w_n\}$ .

*Step 2:* Preprocessing

Firstly, we pre-process each document ( $D$ ) before passing it on to the next step. In pre-processing step the following tasks are performed:

1. Omitting additional characters such as { }, [ ], ...
2. Omitting unessential words such as at, in, on, and, of, ...

3. Finding the roots of the words and the verbs used in the main text.
4. Separating words and omitting repeated words.
5. Identifying the end of the sentences and separating the sentences.
6. Calculating the number of words ( $W$ ) and sentences ( $S$ ) used in the main text.

*Step 3:* Weighting sentences and create similarity matrix

After pre-processing (step 2), the sentences extracted are weighted using TFIDF method in formulas (12) and (13) (Ledeneva et al. 2008). According to formulas (12) and (13),  $freq_{ij}$  represents the number of repetitions of  $i$  word in  $j$  sentence,  $freq_{lj}$  represents  $l$  word in  $j$  sentence,  $\max_i freq_{ij}$  represents the maximum number of  $i$  word repetitions in  $j$  sentence,  $N$  and  $n_i$  represent the number of sentences in an input text and the number of sentences, respectively. Then, a weight of each word in sentence  $w_{ij}$  is calculated, using formula (14).

$$tf_{ij} = \frac{freq_{ij}}{\max_i freq_{ij}} \tag{12}$$

**Table 2** Characteristics of the topics taken from DUC2002 dataset

Number of topics	D061	D065	D067	D070	D073	D075	D079	D085	D0105
Number of documents in each topic	6	8	7	11	6	10	9	9	7

$$idf_{ij} = \log \frac{N}{n_i} \quad (13)$$

$$w_{ij} = tf_{ij} * idf_{ij} \quad (14)$$

Following the sentences weighting, a similarity matrix including extracted sentences and words are created and weighted, using formula (15). In a similarity matrix, sentences are compared based on their keywords and important words. In formula (15),  $\text{sim}(s_i, q)$  is the sentence similarity matrix,  $w_{iq}$  and  $w_{ij}$  are keywords and title weight, and the weight of each word, respectively.

$$\text{sim}(s_i, q) = \frac{\sum_{i=1}^t w_{ij} * w_{iq}}{\sqrt{\sum_i w_{ij}^2 * \sum_{i=1}^t w_{iq}^2}} \quad (15)$$

*Step 4: Processing in MAMHOA*

In this step, as explained in Sect. 4.1, a habitat lattice is created and its parameters are set (Fig. 5). Here, the steps below are followed:

1. Creating a habitat lattice  $H_{\text{size}} \times H_{\text{size}}$  and setting parameters.
2. Assigning similarity matrix including extracted sentences from step 3 to habitats.
3. Evaluating habitat's conditions in order to find the best habitat (Best summarize).
4. Evaluating new habitat repeatedly if conditions are not met.

## 5 Experiments and discussions

In this section, to evaluate the performance of the proposed algorithm, several experiments have been conducted whose results are reported below. In Experiment 1, we have compared our proposed algorithm with four non-multi-agent algorithms for 200-word and 400-word documents summarization. In the experiment 2, we have compared the proposed algorithm with several Baseline and state-of-the-art methods on 100-word documents published in the literature on DUC 2002. For all experiments, each reported value is obtained by averaging over 200 runs. In all of the tables of our experiments, the best value of each row or each column is shown in bold.

### 5.1 DUC 2002 datasets

DUC 2002 is a standard datasets of documents for evaluation in the field of summarization, created by NIST (National Institute of Standards and Technology). In all of the experiments, we chose 9 DUC 2002 topics (D061, D065, D067, D070, D073, D075, D079, D085, and D0105). Totally, 73 documents were chosen for tests. Table 2 presents the characteristics of the topics in DUC2002 dataset, which are used in Experiment 1 and Experiment 2.

### 5.2 Evaluation measures

To analyze the performance of the proposed algorithm we used Recall-Oriented Understudy for Gisting Evaluation (ROUGE) (Lin 2004). This software includes different packages, which determine the similarity between a summary generated by computer and a summary generated by human. It has been shown that ROUGE is very effective for measuring document summarization. The ROUGE metrics are defined according to different Ns and different strategies, such as ROUGE-N (N: 1, 2), ROUGE-L, ROUGE-S and ROUGE-SU. ROUGE-1 is used to compare the overlap between the system summary and the manual summaries created by human. ROUGE-2 is the measure between the candidate summary and final summary. ROUGE-L computes the ratio of the length of the summaries' longest common subsequence (LCS) to the length of the reference summary. ROUGE-S uses the overlap ratio of a candidate summary to a set of reference summaries. ROUGE-SU is an extension of ROUGE-S with the addition of unigram as the counting unit. Each of these ROUGE evaluation methods generates three criteria (Recall, Precision, and F-score). In this paper, we report the average of scores generated by ROUGE-1, ROUGE-L and ROUGE-SU values. The criteria of Precision, Recall and F-score are calculated using formulas (16) and (17) and (18) (Abuobieda et al. 2013), respectively. The criterion of Precision in formula (16) is an intersection of extracted summarized sentences and an ideal summary of sentences divided by all the extracted sentences. The criterion of Recall in formula (17) is an intersection of relevant sentences and retrieved sentences divided by all the relevant documents, and F-score in formula (18) is a statistical criterion which is a combination of precision and recall



**Table 3** The criteria average ranges of (Precision, Recall and F-score) based on ROUGE-1 scores. (200-word documents)

Algorithms Topic	BBO			GA			CSOA			BFOA			Proposed		
	P	R	F	P	R	F	P	R	F	P	R	F	P	R	F
d061	0.415	0.639	0.503	0.354	0.575	0.438	0.471	0.480	0.402	0.429	0.362	0.225	0.640	0.501	<b>0.517</b>
d065	0.202	0.632	0.306	0.244	0.623	0.351	0.490	0.291	0.369	0.391	0.417	0.301	0.446	0.594	<b>0.463</b>
d067	0.292	0.640	0.401	0.274	0.560	0.368	0.505	0.371	<b>0.410</b>	0.388	0.428	0.380	0.462	0.302	0.344
d070	0.221	0.650	0.330	0.206	0.602	0.307	0.295	0.270	0.391	0.451	0.381	0.376	0.435	0.586	<b>0.499</b>
d073	0.433	0.511	0.469	0.409	0.358	0.382	0.382	0.243	<b>0.583</b>	0.322	0.511	0.403	0.394	0.179	0.236
d075	0.111	0.560	0.328	0.227	0.657	0.337	0.431	0.298	0.482	0.454	0.439	0.431	0.500	0.533	<b>0.500</b>
d079	0.228	0.747	0.350	0.303	0.713	0.425	0.390	0.489	0.406	0.500	0.488	0.399	0.536	0.601	<b>0.564</b>
d085	0.246	0.705	0.365	0.242	0.660	0.354	0.552	0.511	0.388	0.373	0.370	0.372	0.376	0.469	<b>0.397</b>
d105	0.231	0.648	0.341	0.186	0.643	0.288	0.404	0.200	0.381	0.344	0.297	0.287	0.326	0.700	<b>0.422</b>
Average	0.264	0.636	0.377	0.271	0.599	0.361	0.435	0.435	0.423	0.405	0.410	0.352	0.457	0.496	<b>0.441</b>

**Table 4** The criteria average ranges of (Precision, Recall and F-score) based on ROUGE-L scores. (200-word documents)

Algorithms Topic	BBO			GA			CSOA			BFOA			Proposed		
	P	R	F	P	R	F	P	R	F	P	R	F	P	R	F
d061	0.380	0.585	<b>0.461</b>	0.345	0.560	0.427	0.171	0.537	0.259	0.455	0.419	0.436	0.368	0.547	0.457
d065	0.184	0.578	0.280	0.232	0.591	0.333	0.096	0.550	0.164	0.360	0.300	0.330	0.160	0.560	<b>0.360</b>
d067	0.278	0.610	<b>0.382</b>	0.245	0.500	0.328	0.137	0.528	0.217	0.230	0.299	0.260	0.246	0.509	0.378
d070	0.213	0.626	0.318	0.196	0.574	0.293	0.089	0.650	0.157	0.300	0.400	0.350	0.192	0.710	<b>0.451</b>
d073	0.421	0.497	<b>0.456</b>	0.398	0.348	0.372	0.186	0.357	0.244	0.337	0.297	0.329	0.346	0.448	0.397
d075	0.205	0.495	0.290	0.209	0.606	0.311	0.086	0.495	0.147	0.349	0.401	0.386	0.196	0.467	<b>0.331</b>
d079	0.225	0.737	0.345	0.290	0.684	<b>0.408</b>	0.121	0.541	0.198	0.351	0.441	0.391	0.188	0.587	0.387
d085	0.231	0.660	0.342	0.227	0.620	0.333	0.101	0.533	0.170	0.274	0.325	0.297	0.202	0.563	<b>0.382</b>
d105	0.219	0.614	0.323	0.169	0.585	0.262	0.083	0.590	0.145	0.294	0.404	0.340	0.204	0.590	<b>0.397</b>
Average	0.261	0.600	0.355	0.256	0.563	0.340	0.204	0.531	0.189	0.327	0.365	0.308	0.233	0.553	<b>0.393</b>

criteria and shows the score for final selected sentences in a summary.

$$\text{Precision} = \frac{\text{RelevantSentences} \cap \text{RetrievedSentences}}{\text{RetrievedSentences}} \quad (16)$$

$$\text{Recall} = \frac{\text{RelevantSentences} \cap \text{RetrievedSentences}}{\text{RelevantSentences}} \quad (17)$$

$$F - \text{score} = \frac{2 * \text{precision} * \text{recall}}{\text{precision} + \text{recall}} \quad (18)$$

### 5.3 Experiment 1: Comparison of proposed algorithm with the non-multi-agent algorithms

In this experiment, we compare the performance of the proposed algorithm with four non-multi-agent optimization algorithms i.e. BBO (Mirshojaee et al. 2017), CSOA (Mirshojaei and Masoomi 2015), BFOA (Asgari and Masoumi 2013) and

GA (Fattah and Ren 2009). Tables 3, 4, and 5 show the analytical results of Precision (P), Recall (R), and F-score (F) criteria for 200-word text summarization using ROUGE-1, ROUGE-L and ROUGE-SU. In experiment 1, we evaluate the results of F-score criteria of our algorithm with other algorithms.

Figure 7 and graph in Fig. 8 compare the average of F-score criteria of the proposed algorithm with those obtained by other optimization Algorithms. As the results show, the summarization F-score criterion for 200-word document has optimized.

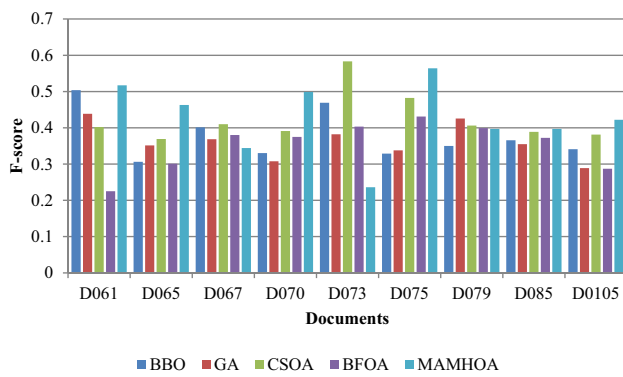
In Tables 4 and 5, the comparison of the Precision, Recall and F-score criteria are shown by ROUGE-L and ROUGE-SU.

Below, Tables 6, 7 and 8 show the analytical results of Precision (P), Recall (R), and F-score (F) criteria for 400-word documents using ROUGE-1, ROUGE-L and ROUGE-SU.

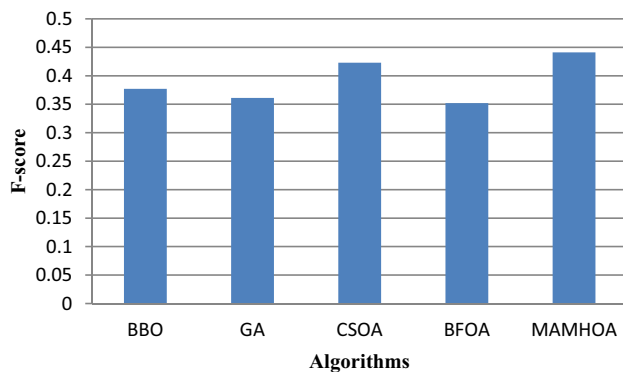
Figure 9 and graph in Fig. 10 show that the proposed algorithm has an optimum performance compared to CSOA and BFOA. According to the graph in Fig. 10, the

**Table 5** The criteria average ranges of Precision, Recall and F-score based on ROUGE-SU scores. (200-word documents)

Algorithms	BBO			GA			CSOA			BFOA			Proposed		
Topic	P	R	F	P	R	F	P	R	F	P	R	F	P	R	F
d061	0.171	0.404	<b>0.241</b>	0.129	0.339	0.187	0.031	0.304	0.167	0.192	0.163	0.176	0.072	0.321	0.196
d065	0.039	0.390	0.072	0.132	0.291	0.211	0.010	0.336	0.173	0.191	0.201	0.196	0.094	0.357	<b>0.225</b>
d067	0.076	0.366	0.126	0.063	0.265	<b>0.164</b>	0.018	0.271	0.144	0.052	0.088	0.065	0.032	0.275	0.153
d070	0.047	0.408	0.085	0.082	0.362	0.076	0.008	0.358	0.183	0.194	0.209	0.201	0.071	0.350	<b>0.210</b>
d073	0.172	0.240	<b>0.200</b>	0.156	0.119	0.135	0.032	0.211	0.121	0.166	0.109	0.132	0.064	0.216	0.140
d075	0.050	0.290	0.085	0.084	0.386	0.082	0.009	0.270	0.139	0.122	0.192	0.149	0.024	0.274	<b>0.149</b>
d079	0.052	0.560	0.096	0.091	0.503	0.154	0.016	0.330	0.173	0.157	0.201	0.151	0.018	0.367	<b>0.192</b>
d085	0.058	0.478	0.104	0.054	0.404	0.096	0.010	0.229	0.119	0.158	0.221	<b>0.184</b>	0.020	0.311	0.165
d105	0.048	0.384	0.086	0.032	0.393	0.060	0.007	0.357	0.182	0.090	0.170	0.117	0.021	0.345	<b>0.183</b>
Average	0.079	0.391	0.121	0.091	0.340	0.147	0.025	0.296	0.155	0.146	0.172	0.152	0.046	0.312	<b>0.179</b>



**Fig. 7** Comparison between F-score criteria of proposed algorithm and those of other algorithms by ROUGE-1. (200-word documents)



**Fig. 8** Comparison between the averages of F-score criteria of the proposed algorithm and those obtained by other optimization algorithms. (200-word documents)

performance of the proposed algorithm is near that of GA and BBO.

Table 9 and graphs in Figs. 11 and 12 show the results of comparison between the average of F-scores obtained

for ROUGE-1, ROUGE-L and ROUGE-SU on 200-word and 400-word documents.

### 5.3.1 Discussion 1

At experiment 1, we discuss four non-multi-agent optimization algorithms including BBO, GA, CSOA and BFOA, which are, cited approaches in the literature. These algorithms are selected for comparative analysis on DUC 2002 documents. The analytical values of F-score criteria are generated using ROUGE Software. The Tables 3, 4, 5, 6, 7, and 8 shows the results of F-score criteria obtained for ROUGE-1, ROUGE-SU, and ROUGE-L. The Table 9 and graphs in the Figs. 11 and 12 show the averages of F-score values are broken down into ROUGE-1, ROUGE-L and ROUGE-SU for 200-word and 400-word documents, respectively.

From the results obtained from ROUGE-1, ROUGE-SU, and ROUGE-L the following points can be made.

#### 5.3.1.1 200-word documents summarization

1. Based on the results of ROUGE-1 (Table 3), the f-score criteria extracted using the proposed algorithm for documents d061, d065, d070, d075, d079, d085 and d105 are more optimal than those obtained by other algorithms. Given the criterion of ROUGE-1, there is more overlap between the extracted sentences of the summary and the sentences of the original text. CSOA performs better on documents d067 and d073.
2. Based on the results of ROUGE-L (Table 4), the f-score criteria extracted using the proposed algorithm for documents d065, d070, d075, d085 and d105 are more optimal than those obtained by other algorithms. Given the criterion of ROUGE-L, the ratio of the length of the summaries' longest common subsequence (LCS) to the length of the reference summary has optimized. Algo-

**Table 6** The criteria average ranges of (Precision, Recall and F-score) based on ROUGE-1 scores. (400-word documents)

Algorithms	BBO			GA			CSOA			BFOA			Proposed		
	P	R	F	P	R	F	P	R	F	P	R	F	P	R	F
d061	0.641	0.490	<b>0.555</b>	0.571	0.461	0.510	0.410	0.631	0.497	0.461	0.428	0.435	0.665	0.419	0.514
d065	0.329	0.594	0.423	0.369	0.542	0.439	0.283	0.612	0.387	0.396	0.479	0.379	0.420	0.568	<b>0.483</b>
d067	0.349	0.600	0.437	0.426	0.415	0.420	0.502	0.431	<b>0.464</b>	0.456	0.446	0.441	0.451	0.291	0.354
d070	0.375	0.674	0.482	0.395	0.703	0.506	0.331	0.813	0.471	0.476	0.406	0.417	0.437	0.598	<b>0.506</b>
d073	0.590	0.362	0.449	0.622	0.282	0.388	0.455	0.480	0.467	0.356	0.586	<b>0.546</b>	0.466	0.169	0.248
d075	0.451	0.550	0.496	0.430	0.629	0.511	0.324	0.747	0.452	0.467	0.487	0.489	0.509	0.512	<b>0.511</b>
d079	0.412	0.652	0.505	0.509	0.579	0.542	0.280	0.590	0.380	0.511	0.501	0.411	0.546	0.622	<b>0.581</b>
d085	0.387	0.553	<b>0.455</b>	0.365	0.498	0.422	0.287	0.604	0.389	0.400	0.390	0.416	0.386	0.403	0.400
d105	0.431	0.664	<b>0.523</b>	0.334	0.635	0.438	0.298	0.731	0.423	0.375	0.395	0.391	0.346	0.689	0.460
Average	0.440	0.571	<b>0.480</b>	0.446	0.527	0.464	0.352	0.626	0.436	0.433	0.457	0.436	0.469	0.474	0.451

**Table 7** The criteria average ranges of (Precision, Recall and F-score) based on ROUGE-L scores. (400-word documents)

Algorithms	BBO			GA			CSOA			BFOA			Proposed		
	P	R	F	P	R	F	P	R	F	P	R	F	P	R	F
d061	0.612	0.468	<b>0.530</b>	0.571	0.461	0.510	0.369	0.567	0.447	0.387	0.582	0.465	0.646	0.407	0.500
d065	0.299	0.540	0.385	0.369	0.542	0.439	0.264	0.571	0.361	0.270	0.452	0.351	0.397	0.537	<b>0.457</b>
d067	0.307	0.586	0.401	0.426	0.415	0.420	0.480	0.412	<b>0.441</b>	0.302	0.369	0.335	0.422	0.272	0.330
d070	0.364	0.654	0.468	0.395	0.703	0.506	0.326	0.801	0.464	0.367	0.450	0.408	0.421	0.575	<b>0.486</b>
d073	0.559	0.343	0.425	0.622	0.282	0.388	0.398	0.420	0.408	0.384	0.478	<b>0.431</b>	0.426	0.154	0.226
d075	0.432	0.526	0.475	0.430	0.629	<b>0.511</b>	0.317	0.730	0.442	0.257	0.400	0.328	0.488	0.491	0.490
d079	0.393	0.622	0.481	0.509	0.579	0.542	0.275	0.578	0.373	0.314	0.478	0.396	0.523	0.596	<b>0.557</b>
d085	0.369	0.528	<b>0.435</b>	0.365	0.498	0.422	0.277	0.560	0.370	0.281	0.573	0.377	0.359	0.375	0.367
d105	0.426	0.656	<b>0.516</b>	0.334	0.635	0.438	0.295	0.723	0.419	0.280	0.510	0.395	0.339	0.675	0.452
Average	0.417	0.547	0.457	0.446	0.527	<b>0.464</b>	0.333	0.595	0.413	0.315	0.476	0.387	0.446	0.453	0.429

**Table 8** The criteria average ranges of (Precision, Recall and F-score) based on ROUGE-SU scores. (400-word documents)

Algorithms	BBO			GA			CSOA			BFOA			Proposed		
	P	R	F	P	R	F	P	R	F	P	R	F	P	R	F
d061	0.404	0.236	<b>0.298</b>	0.312	0.204	0.246	0.155	0.367	0.219	0.155	0.350	0.215	0.393	0.156	0.224
d065	0.104	0.340	0.160	0.129	0.278	0.176	0.079	0.370	0.131	0.084	0.501	0.144	0.173	0.317	<b>0.224</b>
d067	0.105	0.338	0.159	0.163	0.155	0.159	0.201	0.148	0.171	0.250	0.124	0.166	0.191	0.197	<b>0.194</b>
d070	0.139	0.449	0.213	0.152	0.481	0.231	0.104	0.630	0.179	0.138	0.502	0.217	0.189	0.354	<b>0.247</b>
d073	0.338	0.127	0.185	0.366	0.075	0.125	0.188	0.210	0.198	0.240	0.368	<b>0.290</b>	0.227	0.229	0.228
d075	0.181	0.268	0.216	0.161	0.346	0.220	0.098	0.520	0.165	0.122	0.251	0.186	0.233	0.236	<b>0.235</b>
d079	0.163	0.409	0.234	0.249	0.322	0.281	0.080	0.351	0.130	0.105	0.180	0.215	0.277	0.359	<b>0.313</b>
d085	0.146	0.299	<b>0.196</b>	0.129	0.241	0.168	0.075	0.349	0.127	0.084	0.349	0.135	0.144	0.157	0.150
d105	0.188	0.447	<b>0.265</b>	0.109	0.395	0.171	0.089	0.536	0.153	0.095	0.555	0.163	0.122	0.482	0.195
Average	0.196	0.323	0.214	0.196	0.277	0.197	0.118	0.386	0.163	0.141	0.447	0.192	0.216	0.276	<b>0.223</b>

rithm BBO performs better on documents d067, d073 and d061 and GA performs better on document d079.

3. Based on the results of ROUGE-SU (Table 5), the f-score criteria extracted using the proposed algorithm for documents d070, d065, d075, d079 and d105 are

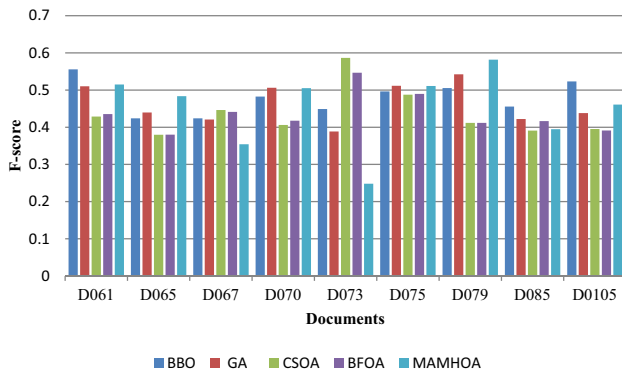


Fig. 9 Comparison between the proposed algorithm and other algorithms by ROUGE-1 in terms of F-score. (400-word documents)

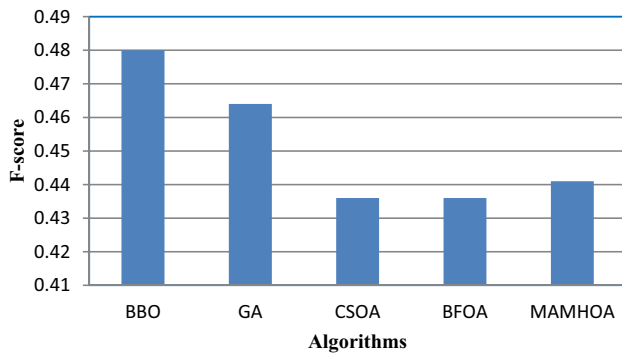


Fig. 10 Comparison between the averages of F-score criteria obtained by the proposed algorithm and those obtained by other optimization algorithms. (400-word documents)

more optimal than those obtained by other algorithms. Given the criterion of ROUGE-SU, word redundancy and repeated words have decreased in the summary. Algorithm BBO performs better on documents d073 and d061. GA performs better on document d067 and BFOA performs better on document d085.

Table 9 Comparison between the averages of obtained F-scores from the mentioned algorithms and those from the proposed algorithm

Algorithms	ROUGE-1		ROUGE-L		ROUGE-SU	
	400-word	200-word	400-word	200-word	400-word	200-word
BBO	<b>0.48</b>	0.377	0.457	0.355	0.214	0.121
GA	0.464	0.361	<b>0.464</b>	0.340	0.197	0.147
BFOA	0.436	0.352	0.387	0.308	0.152	0.192
CSOA	0.436	0.423	0.413	0.189	0.155	0.163
MAMHOA	0.451	<b>0.441</b>	0.429	<b>0.393</b>	<b>0.223</b>	<b>0.179</b>

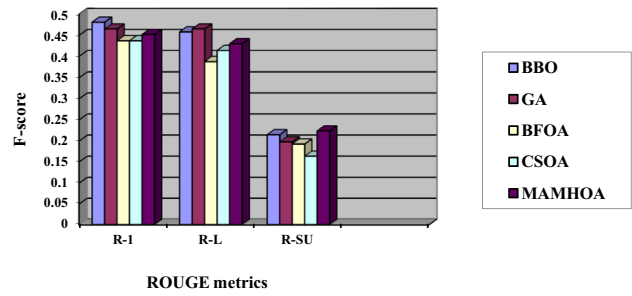


Fig. 11 Comparison between the averages of F-score criteria obtained from the proposed algorithm and those obtained by other optimization algorithms. (400-word documents)

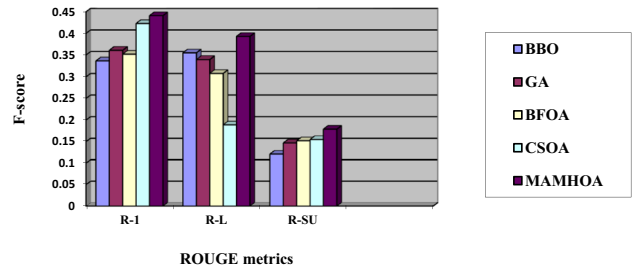


Fig. 12 Comparison between the averages of F-score criteria obtained from the proposed algorithm and those obtained by other optimization algorithms. (200-word documents)

### 5.3.1.2 400-word documents summarization

- Based on the results of ROUGE-1 (Table 6), the F-score criteria extracted using the proposed algorithm for documents d070, d065, d079, and d075 are more optimal than those obtained by other algorithms. Given the criterion of ROUGE-1, there is more overlap between the extracted sentences of the summary and the sentences of the original text. BFOA performs better on document d073. CSOA performs better on document d067. Algorithm BBO performs better on documents d061, d085 and d105.

**Table 10** Baseline extractive summarization algorithms

Baseline methods	Year	Comments
Luhn	1958	Selects the leading sentences in the document to form a summary. It is often used as an official baseline of DUC
TextRank	2004	Is a very fast algorithm for text summarization based on statistical features such as the frequency of words and the relative positioning of each sentence in the document
LexRank	2004	Is a graph-based technique apply the concept of eigenvector centrality to the graph representation of sentences to estimate the importance of each sentence in the document
LSA	2004	Is a topic-based approach that tries to find salient sentences in the document by applying Singular Value Decomposition (SVD) over document matrix
SumBasic	2005	Is a topic-based approach that tries to find salient sentences in the document by applying Singular Value Decomposition (SVD) over document matrix
KLSum	2009	Is a greedy search approximation algorithm that uses a frequency-based sentence selection component with a component to re-weight the word probabilities to minimize redundancy

- Based on the results of ROUGE-L (Table 7), the f-score criteria extracted using the proposed algorithm for documents d065, d070 and d079 are more optimal than those obtained by other algorithms. Given the criterion of ROUGE-L, the ratio of the length of the summaries' longest common subsequence (LCS) to the length of the reference summary has optimized. Algorithm BBO performs better on documents d061, d105 and d085. GA performs better on document d075. CSOA performs better on document d067 and BFOA performs better on document d073.
- Based on the results of ROUGE-SU (Table 8), the f-score criteria extracted using the proposed algorithm for documents d070, d065, d075, d079 and d067 are more optimal than those obtained by other algorithms. Given the criterion of ROUGE-SU, word redundancy and repeated words have decreased in the summary. Algorithm BBO performs better on documents d061, d085 and d105. BFOA performs better on document d073.

In sum, based on the above discussion as well as Table 9 and graphs in Figs. 11 and 12, the proposed algorithm is generally more efficient in the summarization of 200-word documents compared to that of 400-word documents due to the following: more overlap between the summary and the original text, smaller ratio of the length of the summaries' longest common subsequence to the length of the reference summary, and a decrease in the redundancy of words.

## 5.4 Experiment 2: comparison of proposed algorithm with the baseline and state-of-the-art benchmarks

In experiment 2, the efficiency of the proposed algorithm is compared to that of the algorithms discussed in literature review including baseline and state-of-the-art algorithms. In this experiment, the proposed algorithm was applied to 100-word documents. The results of the proposed algorithm were compared to those of other mentioned algorithms in terms of Recall criterion.

### 5.4.1 Baseline and state-of-the-art extractive summarization algorithms

In this section, we compare our algorithm with the several widely used baseline approaches (Table 10), and seven state-of-the-art algorithms published in literature on the dataset DUC 2002 (Table 11). We compared our work with baseline and state-of-the-art algorithms. The methods are selected based on state-of-the-art accuracy reported by the, which are the most cited approaches in the literature (i.e. between 2010 and 2017). A brief description of the baseline approaches and state-of-the-art are shown in Tables 10 and 11.

### 5.4.2 Comparative analysis on DUC 2002 dataset

Table 12 show the average of Recall obtained for the summarization of 100-word datasets of DUC 2002. It can be observed that on these dataset, MAMHOA reports the highest averages for ROUGE-1 (51.0), ROUGE-SU (24.3) and ROUGE-L (47.2) on 100-word documents.

**Table 11** Seven state-of-the-art extractive summarization algorithms

State-of-the-art methods	Year	Comments
ILP	2010	Encodes the grammaticality constraints across phrase dependencies using integer linear programming
Egraph	2015	Is considered as an entity graph-based system, which makes use of a bipartite graph of sentence and entity nodes for document representation
URANK	2010	Is a unified rank methodology based on graph model. It is able to handle single-document and multi-document summarization simultaneously
Tgraph	2015	Is an unsupervised graph-based text summarization algorithm that uses Latent Dirichlet Allocation (LDA) for topic modeling.
NN-SE	2016	Which is a neural summarization model for sentence extraction
CoRank	2017	Is a graph-based unsupervised extractive text summarization approach which involves combining word-sentence relationship with the graph-based ranking model
SummaRuNNer	2016	Is a recent extractive text summarization algorithm, which is based on RNNs

**Table 12** ROUGE recall scores of MAMHOA approach for different topics of DUC 2002

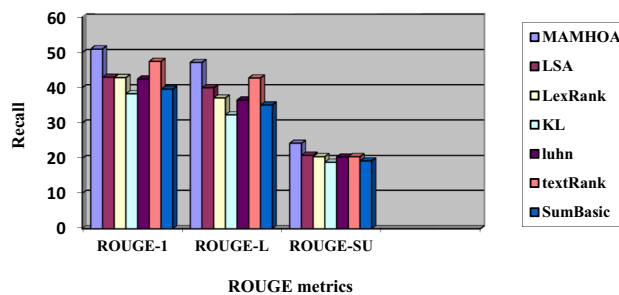
ROUGE	D061	D065	D067	D070	D073	D075	D079	D085	D0105	Average
ROUGE-1	59.0	55.4	37.2	62.4	51.5	51.0	50.7	47.7	44.5	<b>51.0</b>
ROUGE-SU	32.8	29.7	11.8	35.0	19.7	24.9	23.5	20.8	20.6	<b>24.3</b>
ROUGE-L	54.2	50.4	37.2	57.4	46.4	47.2	48.5	44.0	39.6	<b>47.2</b>

**Table 13** Comparison between the averages of Recall criteria obtained from the MAMHOA with those of Baseline and state-of-the-art approaches

Algorithm	ROUGE-1	ROUGE-SU	ROUGE-L
<b>Baseline methods</b>			
DUC 2002	48	–	–
Luhn	42.5	20.4	36.5
TextRank	47	21.7	42.8
LexRank	42.9	20.5	37.1
LSA	43	20.9	40
KLSum	38.3	18.9	32.3
SumBasic	39.6	19.3	35.1
<b>State-of-the-art methods</b>			
ILP	45.4	–	42.8
Egraph + coh	47.9	24.2	–
Tgraph + coh	48.1	23	–
URANK	48.5	–	–
NN-SE	47.4	–	–
CoRank	<b>52.6</b>	–	43.5
SummaRuNNer	47.4	–	14.7
<b>Proposed</b>			
MAMHOA	51	<b>24.3</b>	<b>47.2</b>

**5.4.3 Comparative analysis on DUC 2002 benchmark**

The averages of Recall criteria obtained from the MAMHOA are compared to Recall criteria obtained for Baseline and state-of-the-art approaches.



**Fig. 13** Comparison between the averages of Recall criterion obtained from the proposed algorithm with Baseline algorithms. (100-word documents)

Table 13 and graph in Fig. 13 show the comparative results of efficiency for the proposed algorithm and baseline and state-of-the-art algorithms in terms of Recall, using ROUGE-1, ROUGE-L, and ROUGE-SU.

**5.4.4 Discussion 2**

The results in Table 13 show ROUGE-1 score of 51.0, ROUGE-SU score of 24.3 and ROUGE-L score of 47.2 on this dataset. As a result, our proposed algorithm (MAMHOA) has a better performance than many of the existing state-of-the-art text summarizers on DUC 2002 dataset with respect to ROUGE-1, ROUGE-SU and ROUGE-L scores such as ILP, graph-based approaches, Tgraph, Egraph, URANK, and even those, which are based on supervised learning such as SummaRuNNer and NN-SE. Our approach

cannot surpass the ROUGE-1 score of a recently proposed CoRank approach, though our ROUGE-SU and ROUGE-L scores, which are better evaluation measures than ROUGE-1, are far better than the CoRank method.

The graph in Fig. 13 Show the comparison between the averages of Recall criterion obtained from the proposed algorithm with Baseline algorithms for 100-word documents. As shown in Fig. 13 and Table 13, the performances of proposed algorithm is better than Baseline and stat-of-the-art mentioned algorithms.

## 6 Conclusion

In this work, we have discussed an optimization algorithm named MAMHOA with two strategies for extractive text summarization to find the best or near the best optimum summary. The proposed algorithm is a hybrid of a biogeography meta-heuristic algorithm and multi-agent systems. The proposed algorithm finds the best or near the best summary by creating a habitat lattice and comparing them to find the best conditions. The proposed algorithm was used for an extractive document text summarization issue. Our algorithm was compared with several non-multi-agent optimization algorithms, baseline and state-of-the-art algorithms using DUC2002 datasets. The 100-word and 200-word and 400-word documents were selected for summarization.

In the first experiment, we compared performance of our algorithm with that of some non-multi-agent algorithm on 200-word and 400-word documents and F-score results were obtained. In the second experiment we compared our algorithm with Baseline and state-of-the-art algorithms with 100-word documents and the Recall results were obtained. The experimental results were analyzed, using ROUGE software. Generally, the results show that the proposed algorithm has a better performance than that of other mentioned algorithms so that it can be used for summarizing.

## References

- Abuobieda A, Salim N, Kumar Y et al (2013) An improved evolutionary algorithm for extractive text summarization. *Intell Inf Database Syst LNAI* 7803:78–89
- Amancio JV, Tohalino R (2018) Extractive multi-document summarization using multi-layer networks. *Physica A* 503:526–539
- Asgari H, Masoumi M (2013) Provide a method to improve the performance of text summarization using bacterial foraging optimization algorithm. Seventh Iran data mining conference
- Chen X, Tianfield H (2016) Biogeography-based optimization with covariance matrix based migration. *Appl Soft Comput* 45:71–85
- Cheng J, Lapata M (2016) Neural summarization by extracting sentences and words. In: Proceedings of the 54th annual meeting of the association for computational linguistics, Berlin, Germany, pp 484–494
- Fang C, Mu D, Deng Z, Wu Z (2017) Word -sentence co-ranking for automatic extractive text summarization. *Expert Syst Appl* 72:189–195
- Fattah A, Ren F (2009) GA, MR, FFNN, PNN and GMM based models for automatic text summarization. *Comput Speech Lang* 23:126–144
- Feng Q, Liu S, Zhang J et al (2017) Improved biogeography based optimization with random ring topology and Powell's method. *Appl Math Model* 41:630–649
- Ferreira R, de Souza Cabral L et al (2013) Assessing sentence scoring techniques for extractive text summarization. *Expert Syst Appl* 40(14):5755–5764
- Haghighi A, Vanderwende L (2009) Exploring content models for multi-document summarization. In: Proceedings of human language technologies: the 2009 annual conference of the north American chapter of the association for computational linguistics, Boulder, Colorado, pp 362–370
- Harish G (2015) An efficient biogeography based optimization algorithm for solving reliability optimization problems. *Swarm Evolut Comput* 24:1–10
- Khishe M, Mosavi MR, Kaveh M (2017) Improved migration models of biogeography-based optimization for sonar dataset classification by using neural network. *Appl Acoust* 118:15–29
- Ledeneva Y, Gelbukh A, García-Hernández RA (2008) Terms derived from frequent sequences for extractive text summarization. In: Gelbukh A (ed) Computational linguistics and intelligent text processing. CILing 2008. Lecture notes in computer science, vol 4919. Springer, Berlin, Heidelberg, pp 593–604
- Li Z, Jing L (2016) A multi-agent algorithm for community detection in complex network. *Physical A* 449:336–347
- Lin C-Y (2004) ROUGE: a package for automatic evaluation summaries. In: Proceedings of the workshop on text summarization branches out, Barcelona, Spain, 25–26 July 2004, pp 74–81
- Lohokare MR, Pattnaik SS, Panigrahi BK et al (2013) Accelerated biogeography-based optimization with neighborhood search for optimization. *Appl Soft Comput* 13:2318–2342
- Luhn HP (1958) The automatic creation of literature abstracts. *IBM J Res Dev* 2(2):159–165
- Mihalcea R, Tarau P (2004) TextRank: bringing order into texts. In: Proceedings of the 2004 conference on empirical methods in natural language processing, Barcelona, Spain, pp 404–411
- Mirshojaei SH, Masoumi B, Zeinali E (2017) Biogeography-based optimization algorithm for automatic extractive text summarization. *Int J Ind Eng Prod Res* 28(1):75–84
- Mirshojaei SH, Masoumi B (2015) Text summarization using cuckoo search optimization algorithm. *J Comput Robot* 8(2):19–24
- Nallapati R, Zhou B, dos Santos CN, Gulcehre C, Xiang B (2016) Abstractive text summarization using sequence-to-sequence RNNs and Beyond. In: Proceedings of the SIGNLL Conference on Computational Natural Language Learning, pp 280–290
- Nenkova A, Vanderwende L (2005) The impact of frequency on summarization. Microsoft Research. Technical report MSR-TR-2005-101
- Nenkova A, Maskey S, Liu Y (2011) Automatic summarization. In: Proceedings of the 49th Annual meeting of the Association for Computational Linguistics: Association for computational linguistics vol. 3, pp 1–3:86
- Oufaida H, Nouali O, Blache P (2014) Minimum redundancy and maximum relevance for single and multi- document Arabic text summarization. *J King Saud Univ Comput Inf Sci* 26(4):450–461
- Paraskevopoulos A, Dallas PI, Siakavara K et al (2017) Cognitive radio engine design for IoT using real-coded biogeography-based

- optimization and fuzzy decision making. *Wirel Pers Commun* 97:1813–1833
- Parveen D, Ramsel H-M, Strube M (2015) Topical coherence for graph-based extractive summarization. In: *Proceedings of the 2015 conference on empirical methods in natural language processing*, Lisbon, Portugal, pp 1949–1954
- Radev D, Erkan G (2004) Lexrank: graph-based lexical centrality as salience in text summarization. *J Artif Intell Res* 22:457–479
- Rautray R, Balabantaray RC (2018) An evolutionary framework for multi document summarization using Cuckoo search approach: MDSCSA. *Appl Comput Inf* 14:134–144
- Simon D (2008) Biogeography based optimization. *IEEE Trans Evolut Comput* 12(6):702–713
- Steinberger J, Jezek K (2004) Using latent semantic analysis in text summarization and summary evaluation. In: *Proceedings of 7th international conference ISIM'04*, pp 93–100
- Wan X (2010) Towards a unified approach to simultaneous single-document and multi document summarizations. In: *Proceedings of the 23rd international conference on computational linguistics (Coling 2010)*, Beijing, China, pp 1137–1145
- Woodsend K, Lapata M (2010) Automatic generation of story highlights. In: *Proceedings of the 48th annual meeting of the association for computational linguistics*, Uppsala, Sweden, pp 565–574
- Yang J, Li L (2017) Improved biogeography-based optimization algorithm for mobile robot path planning. *Chin Intell Syst Conf* 2:219–230
- Zhang D, Kanga Q, Chenga J et al (2018) A novel hybrid algorithm based on Biogeography-Based Optimization and Grey Wolf Optimizer. *IEEE Trans Evolut Comput* 67:197–214

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.