Expert Systems With Applications 198 (2022) 116769

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

A multi-objective memetic algorithm for query-oriented text summarization: Medicine texts as a case study

Jesus M. Sanchez-Gomez [a,*], Miguel A. Vega-Rodríguez [a], Carlos J. Pérez [b]

[a] Departamento de Tecnología de Computadores y Comunicaciones, Universidad de Extremadura[1], Campus Universitario s/n, 10003 Cáceres, Spain
[b] Departamento de Matemáticas, Universidad de Extremadura[1], Campus Universitario s/n, 10003 Cáceres, Spain

ARTICLE INFO

Keywords:
Query-oriented summarization
Multi-objective optimization
Memetic algorithm
Recall-oriented understudy for gisting evaluation
Medicine texts

ABSTRACT

Automatic text summarization is a topic of great interest in many fields of knowledge. Particularly, query-oriented extractive multi-document text summarization methods have increased their importance recently, since they can automatically generate a summary according to a query given by the user. One way to address this problem is by multi-objective optimization approaches. In this paper, a memetic algorithm, specifically a Multi-Objective Shuffled Frog-Leaping Algorithm (MOSFLA) has been developed, implemented, and applied to solve the query-oriented extractive multi-document text summarization problem. Experiments have been conducted with datasets from Text Analysis Conference (TAC), and the obtained results have been evaluated with Recall-Oriented Understudy for Gisting Evaluation (ROUGE) metrics. The results have shown that the proposed approach has achieved important improvements with respect to the works of scientific literature. Specifically, 25.41%, 7.13%, and 30.22% of percentage improvements in ROUGE-1, ROUGE-2, and ROUGE-SU4 scores have been respectively reached. In addition, MOSFLA has been applied to medicine texts from the Topically Diverse Query Focus Summarization (TD-QFS) dataset as a case study.

1. Introduction

The number of digital documents published in the World Wide Web and digital libraries has grown extremely in recent years due to the development of information and communication technologies. This information overload makes difficult that users obtain the most useful and relevant information on specific topics. By means of text mining tools, it is possible to extract specific information from a large set of documents (Fan & Bifet, 2013). Particularly, these tools can automatically produce a summary from all the textual information (Hashimi et al., 2015). An automatic summary would fulfill the needs of users, since the volume of information would be considerably reduced while also maintaining the most relevant one.

In the scientific literature, automatic summaries can be generated in several ways. First, text summarization methods can be generic or query-oriented. Generic summarization does not require any information from the user (Alguliev, Aliguliyev, Hajirahimova et al., 2011; Sanchez-Gomez et al., 2018), whereas query-oriented summarization needs some information (specifically, a query) (Huang et al., 2010). The query is usually a narrative sentence that includes a topic of interest given by the user. Besides, an automatic summary may be abstractive or extractive (Wan, 2008). On the one hand, abstractive summaries are made up of words and sentences that may not exist in the original documents. On the other hand, extractive summaries only select subsets of existing text. Text summarization methods can also be single-document or multi-document. Single-document methods reduce the information contained in only one document to a brief presentation, and multi-document methods extract pieces of information from all the documents (Zajic et al., 2008).

In this paper, the focus is centered on the query-oriented extractive multi-document text summarization problem. In recent years, optimization approaches have become very popular in this field due to their robust mathematical formulation, the diversity of adaptable algorithms, and the good results that provide, among other aspects. These approaches can be classified into single-objective or multi-objective optimization. In a single-objective optimization approach, only one objective function is optimized. This objective function includes all the criteria to be weighted. This weighting involves a subjective assignment of the weights, and this influences the final solution of the problem, which is a weakness. On the contrary, multi-objective optimization approaches do not need this subjective assignment, since all the objective functions are simultaneously optimized. This is a great advantage, which does not limit the search to a subjective combination

* Corresponding author.
E-mail addresses: jmsanchezgomez@unex.es (J.M. Sanchez-Gomez), mavega@unex.es (M.A. Vega-Rodríguez), carper@unex.es (C.J. Pérez).
[1] https://ror.org/0174shg90.

https://doi.org/10.1016/j.eswa.2022.116769
Received 18 December 2020; Received in revised form 24 November 2021; Accepted 25 February 2022
Available online 12 March 2022

of weights. For this reason, in this work, a multi-objective optimization approach is developed. Regarding the optimization algorithms, evolutionary algorithms have become really popular, since their stochastic search methods have provided good results in complex optimization problems. Specifically, one of these algorithms is the Shuffled Frog-Leaping Algorithm (SFLA). SFLA is a swarm intelligence algorithm based on population and inspired by the natural memetics of the frog behavior (Eusuff et al., 2006), which has been applied successfully in different real-life problems (see e.g. Elbeltagi et al., 2007; Eusuff & Lansey, 2003; Fang & Wang, 2012; Tang et al., 2020). SFLA is a single-objective optimization algorithm, therefore, its adaptation to the multi-objective context is necessary.

In this paper, a Multi-Objective Shuffled Frog-Leaping Algorithm (MOSFLA) is developed, implemented, and applied to address the query-oriented extractive multi-document text summarization problem. The criteria of query relevance and redundancy reduction have been defined as the two objective functions that have to be optimized. The experiments have been carried out with datasets from Text Analysis Conference (TAC) (McNamee & Dang, 2009). The results have been evaluated by using Recall-Oriented Understudy for Gisting Evaluation (ROUGE) metrics (Lin, 2004). In addition, MOSFLA has also been applied to medicine texts from the Topically Diverse Query Focus Summarization (TD-QFS) (2016) dataset. Therefore, the main contributions of this paper can be summarized as:

- The query-oriented extractive multi-document text summarization problem has been formulated as a multi-objective optimization problem involving two objective functions: query relevance and redundancy reduction.
- MOSFLA, a multi-objective memetic algorithm based on swarm intelligence, has been developed and adapted to solve this problem for the first time.
- In addition to TAC datasets, MOSFLA has been also applied to a real-world case in medicine texts from the TD-QFS dataset.

The remainder of this paper is the following. In Section 2, the related work is presented. Section 3 formulates the query-oriented extractive multi-document text summarization problem as a multi-objective optimization problem. In Section 4, the Multi-Objective Shuffled Frog-Leaping Algorithm is described as well as its operators. In Section 5, the datasets, the evaluation metrics, the results obtained, and their statistical analysis are presented, including the comparisons with other approaches from the scientific literature. Section 6 contains the application of MOSFLA to medicine texts from the TD-QFS dataset as a case study. Finally, in Section 7, the conclusions and the future research are included.

2. Related work

In this section, a review of the approaches used for the query-oriented extractive multi-document text summarization problem is presented.

In Li and Li (2013), a novel sentence feature-based Bayesian model based on supervised Latent Dirichlet Allocation (S-sLDA) was proposed. This proposal combined feature-based supervised methods and topic models, transforming the problem of finding optimum feature weights into an optimization problem. Some widely used models for query-oriented summarization were compared, such as LexRank (Erkan & Radev, 2004), MEAD (Radev et al., 2004), Manifold (Wan et al., 2007), and SVM (Li et al., 2009). LexRank was introduced in Erkan and Radev (2004), and it consists of a stochastic graph-based method for computing the relative importance of units of text. This method was considered for computing the sentence relevance based on the concept of eigenvector centrality in a graph-based representation. Radev et al. (2004) presented the multi-document summarizer MEAD, which is a method that produces summaries by using cluster centroids generated by a topic detection and a tracking system. Two techniques were described: cluster-based relative utility, which is a centroid-based summarizer, and cross-sentence informational subsumption, which is an evaluation scheme based on sentence utility and subsumption. Wan et al. (2007) presented Manifold, a novel extractive approach based on manifold-ranking of sentences. It used the manifold-ranking process to compute the manifold-ranking score for every sentence, using then a greedy algorithm to penalize sentences with a high degree of overlapping. The Support Vector Machine (SVM) method was used in Li et al. (2009). The summarization problem was formulated as a learning framework, employing the structural Support Vector Machine method and adapting the cutting plane algorithm to solve it.

A two-layer graph-based semi-supervised learning approach based on topic modeling techniques, which extends the standard graph ranking algorithm, was proposed in Li and Li (2014). Two versions of the LDA topic model were described: a word level model (W-LDA) and a sentence level model (S-LDA). To evaluate the performance of these models, several approaches were used for comparison: LexRank, MEAD, Manifold, KL-divergence (Lin et al., 2006), and HS-LDA (Haghighi & Vanderwende, 2009). The KL-divergence model was proposed in Lin et al. (2006), which is an information theoretic approach to automatically evaluate summaries. It was developed by using the KL-divergence based sentence selection strategy. HS-LDA (Haghighi & Vanderwende, 2009) used a hierarchical LDA-style model, a variation of the hierarchical LDA topic model, to represent content specificity as a hierarchy of topic vocabulary distributions.

In Marujo et al. (2015), the extension of the single-document summarization KP-Centrality method to perform multi-document summarization was proposed. Two hierarchical strategies were explored: the single-layer architecture, which aggregates summaries concatenated chronologically ordered, and the waterfall architecture, in which the intermediate summaries are merged. This proposal used the LexRank and MEAD models for comparison purpose. An event detection method based on Fuzzy Fingerprint was proposed in Marujo et al. (2016). That event classification-based approach was supported by two different distributed representations of the text: the skip-ngram model and the bag-of-words model. LexRank and MEAD were also used for comparison. Bossard and Rodrigues (2017) proposed a new generic and directly usable sentence extraction method by considering a system based on an evolutionary algorithm (EA). This optimization approach calculated the distribution probability of tokens in the input documents with the distribution probability in the summaries. Four different optimization models were considered for experimentation, being the bigram distribution (EA BiProb) the one with the best performance. LexRank method was used for comparison.

Finally, in Fors-Isalguez et al. (2018) the query-oriented summarization problem was addressed from a multi-objective optimization point of view. Two different sentence representation models were studied: standard *tf-isf* representation (NSGA-II TF-ISF) and word embedding representation (NSGA-II WE). The algorithm used in Fors-Isalguez et al. (2018) was the Non-dominated Sorting Genetic Algorithm-II (NSGA-II), which has been successfully applied in other real-life multi-objective optimization problems, such as the multi-objective generation expansion planning problem (Murugan et al., 2009), the multi-objective automatic calibration of a physically-based semi-distributed watershed model (Bekele & Nicklow, 2007), and the multi-objective reactive power planning problem (Ramesh et al., 2012).

All the reviewed approaches used the ROUGE metrics in their experimentation. Specifically, ROUGE-1, ROUGE-2, and ROUGE-SU4 scores were evaluated. Besides, all the approaches have been applied in TAC2009 datasets. Therefore, both these three ROUGE scores and TAC2009 datasets will be used to carry out the experiments in this paper for comparative purposes.

3. Problem statement

The query-oriented extractive multi-document text summarization problem is presented in this section. In this field, the most commonly used methods are vector-based word methods. According to them, a sentence is represented as a vector of words, and the similarity measure between two sentences is calculated by using some criterion as, for example, cosine similarity.

3.1. Sentence representation and cosine similarity measure

Firstly, the representation of a sentence as a vector of words is defined. Let $T = \{t_1, t_2, \dots, t_m\}$ represent a set that contains all the different terms from the document collection D, being m the number of terms. Each individual sentence s_i of D can be represented as a vector of m dimensions as $s_i = (w_{i1}, w_{i2}, \dots, w_{im}), i = 1, 2, \dots, n$, where each component refers to the weight of the term t_k in the sentence s_i, and n is the number of sentences. The weight w_{ik} can be calculated by using the *term-frequency inverse-sentence-frequency* (*tf-isf*) scheme (Salton & Buckley, 1988) as indicated in Eq. (1):

$$w_{ik} = tf_{ik} \cdot log(n/n_k), \tag{1}$$

where tf_{ik} counts the times that the term t_k occurs in the sentence s_i, and n_k counts the number of sentences of D in that the term t_k appears.

Now, the cosine similarity measure is described based on the previous sentence representation. This similarity measures the resemblance between two sentences s_i and s_j from the document collection D. It is calculated in Eq. (2):

$$cosim(s_i, s_j) = \frac{\sum_{k=1}^{m} w_{ik} w_{jk}}{\sqrt{\sum_{k=1}^{m} w_{ik}^2 \cdot \sum_{k=1}^{m} w_{jk}^2}}, \quad i, j = 1, 2, \dots, n. \tag{2}$$

3.2. Mathematical formulation of the optimization problem

Once the bases of the problem have been raised, the optimization problem can be formulated. Let the document collection $D = \{d_1, d_2, \dots, d_N\}$ be a set with N documents. The document collection can also be represented as $D = \{s_1, s_2, \dots, s_n\}$, being a set which contains the n sentences from all the documents. The aim of this problem is to produce a summary S that contains some sentences from D (that is, $S \subset D$) taking into account the following points:

- Query relevance. The summary must contain only the sentences which are relevant to the user according to a given query.
- Redundancy reduction. The summary should not contain sentences that are similar among them.
- Length. The summary must have a predetermined length L.

The optimization problem entails the simultaneous optimization of the query relevance and the redundancy reduction. Nevertheless, these two criteria are conflicting to each other. In addition, the summary length constraint also has to be fulfilled. Hence, it seems that the best way to address this problem is through a multi-objective optimization approach.

Now, the objective functions to be optimized are defined, but first it is necessary to define the representation of the solutions. Let the binary variable $x_i \in \{0, 1\}$ consider the presence or absence of the sentence s_i in the summary S, i.e., $x_i = 1$ when $s_i \in S$ and $x_i = 0$ when $s_i \notin S$. Thus, the representation of a solution is given by the vector $X = (x_1, x_2, \dots, x_n)$.

The first objective function refers to the query relevance criterion: $\Phi_{query\_rel}(X)$. The query relevance is defined as the cosine similarity between each sentence in the summary $s_i \in S$ and the query vector $Q = (q_1, q_2, \dots, q_m)$. Q represents the query given by the user as a sentence, and its weights q_k are calculated in the same way as was explained in Eq. (1). Therefore, the objective function in Eq. (3) should be maximized:

$$\Phi_{query\_rel}(X) = \sum_{i=1}^{n} cosim(s_i, Q) \cdot x_i. \tag{3}$$

The second objective function concerns the redundancy reduction criterion: $\Phi_{redun\_red}(X)$. The redundancy reduction expresses that the cosine similarity between each pair of sentences of the summary $s_i, s_j \in S$ should be reduced, so it is equivalent to maximize the objective function in Eq. (4):

$$\Phi_{redun\_red}(X) = \frac{1}{\sum_{i=1}^{n-1} \sum_{j=i+1}^{n} cosim(s_i, s_j) \cdot x_i x_j}. \tag{4}$$

Finally, the query-oriented extractive multi-document text summarization problem addressed from the multi-objective optimization point of view is formulated in Eqs. (5) and (6):

$$\max \ \Phi(X) = \{\Phi_{query\_rel}(X), \Phi_{redun\_red}(X)\}, \tag{5}$$

$$\text{subject to } L - \varepsilon \leq \sum_{i=1}^{n} l_i \cdot x_i \leq L + \varepsilon, \tag{6}$$

where l_i is the length of the sentence s_i and ε is the tolerance for the summary length constraint. Eq. (6) is calculated as in Alguliev, Aliguliyev, and Mehdiyev (2011). The value of the tolerance ε is defined as the difference between the length of the longest sentence and the length of the shortest sentence from the document collection. It is calculated as:

$$\varepsilon = \max_{i=1,2,\dots,n} l_i - \min_{i=1,2,\dots,n} l_i. \tag{7}$$

4. Multi-objective shuffled frog-leaping algorithm

In this section, the Multi-Objective Shuffled Frog-Leaping Algorithm is presented. Firstly, the basic algorithm (SFLA) is described. Then, the preprocessing steps are defined. And finally, the main steps of MOSFLA and its main operators are explained.

4.1. Basic algorithm

The Shuffled Frog-Leaping Algorithm was proposed in Eusuff et al. (2006) for solving optimization problems. SFLA was developed as a population-based cooperative search algorithm, and it was inspired by natural memetics of the frog behavior. The algorithm consists of a population of virtual frogs partitioned into different memeplexes, where they interact with each other. Every virtual frog represents a solution to the problem. The virtual frogs are shuffled and then reorganized into new memeplexes regularly in order to ensure the global search. Moreover, some virtual frogs are randomly generated and replace others in the population to give the chance to produce new explorations. The main steps of SFLA are described in Algorithm 1.

Algorithm 1 Basic SFLA pseudocode.

```
Population ← init_population(pop_size)
Population ← calculate_fitness(Population)
Population ← sort_by_fitness(Population)
while not stop_criteria do
    Memeplexes ← divide_pop_into_memes(Population, memes_num)
    Memeplexes ← local_search(Memeplexes, memes_num, improvs_max)
    Population ← combine_evolved_memeplexes(Memeplexes)
    Population ← calculate_fitness(Population)
    Population ← sort_by_fitness(Population)
end while
```

The basic algorithm starts creating a random initial population $Population$ with a number of frogs pop_{size}. After that, the fitness value

of each individual frog is calculated, and then the entire population is ordered in descending way according to the calculated fitnesses.

Now, the following tasks are performed until the stopping criteria are met. First, the population (set of solutions) is partitioned into a number of memeplexes equal to $memes_{num}$, containing each one of them a number of frogs (solutions) equal to $frogs_{num}$, that is, $pop_{size} = memes_{num} \times frogs_{num}$. The ordered frogs are distributed among the memeplexes in a shuffled way, i.e., the first frog goes to the first memeplex, the second one goes to the second memeplex, and, in general, the frog from position $memes_{num}$ goes to the last memeplex (the number $memes_{num}$). Then, the frog from position $memes_{num} + 1$ goes to the first memeplex again, and so on. Therefore, the shuffling process assures that all the memeplexes have frogs (solutions) of all the qualities, that is, solutions with the best, medium, and worst fitnesses from the population. After shuffling, each individual frog is contained as a whole in its corresponding memeplex.

Secondly, the local search is carried out within each memeplex (subset of solutions). This task consists of the improvement of the worst frog (the one with the worst fitness value in the memeplex) during a maximum number of improvements $improvs_{max}$, which is replaced by a better mutated frog or a frog randomly generated.

After the local search is finished, the resulting evolved memeplexes are combined into the population. Finally, the fitness values are calculated again for all the frogs and the population is ordered in descending way according to the new fitnesses, finishing a cycle. A more detailed explanation of the basic algorithm can be found in Eusuff et al. (2006).

SFLA has been successfully applied in different real-life optimization problems. Some problems are, for example, optimization of water distribution network design (Eusuff & Lansey, 2003), project management (Elbeltagi et al., 2007), resource-constrained project scheduling (Fang & Wang, 2012), or influence maximization in social networks (Tang et al., 2020).

4.2. Preprocessing

Before the execution of MOSFLA, some preprocessing steps need to be carried out with the documents from the collection D:

1. Segmentation. All the sentences from the document collection have to be extracted in a separate way, delimiting their beginning and ending.
2. Tokenization. All the words of the sentences are separated. Moreover, exclamations, interrogations, punctuations, and other marks are removed from the sentences.
3. Stop words removal. The words with no meaning such as prepositions, conjunctions, articles, and others are deleted from the sentences. The stop words list used is provided by the ROUGE package, and it includes a total of 598 words (Li, 2020).
4. Stemming. Finally, the root of each remaining word is extracted with the Porter stemming algorithm (Porter, 2020), so the words that share a common lexical root will be processed as a single term.

4.3. Steps of MOSFLA

The algorithm implemented in this paper consists of the adaptation of SFLA to a multi-objective optimization approach with some improvement. The steps of MOSFLA are detailed in Algorithm 2 and explained below.

Algorithm 2 MOSFLA pseudocode.

```
1:  NDS_file ← ∅
2:  Population ← init_population(pop_size)
3:  Population ← calculate_objective_functions(Population, pop_size)
4:  Population ← sort_by_rank_and_crowding(Population, pop_size)
5:  for cycle = 1 to cycles_max do
6:    Memeplexes ← divide_pop_into_memes(Population, memes_num)
7:    X_bestG ← select_best_global_solution(Memeplexes)
8:    for m = 1 to memes_num do
9:      for i = 1 to improvs_max do
10:       X_bestL ← select_best_local_solution(Memeplexes[m])
11:       X_worstL ← select_worst_local_solution(Memeplexes[m])
12:       save_worst_local_solution(Population, X_worstL)
13:       X_new ← mutate_solution(X_bestL, p_m)
14:       if X_new ≻ X_worstL then
15:         save_solution(Memeplexes[m], X_new)
16:       else
17:         X_new ← mutate_solution(X_bestG, p_m)
18:         if X_new ≻ X_worstL then
19:           save_solution(Memeplexes[m], X_new)
20:         else
21:           X_new ← random_solution()
22:           save_solution(Memeplexes[m], X_new)
23:         end if
24:       end if
25:       Memeplexes[m] ← sort_by_dominance(Memeplexes[m])
26:     end for
27:   end for
28:   Population ← combine_evolved_memeplexes(Memeplexes)
29:   Population ← calculate_objective_functions(Population, pop_size * 2)
30:   Population ← sort_by_rank_and_crowding(Population, pop_size * 2)
31:   save_nondominated_solutions(Population, NDS_file)
32: end for
```

In the first place, the file that will store the non-dominated solutions, NDS_file, is initialized. After that, the initial population, $Population$, is randomly generated with pop_{size} frogs (solutions). Then, the values of the objective functions for every solution are calculated, and the population is ordered according to two multi-objective metrics: rank and crowding distance (Deb et al., 2002). The rank indicates the layer of the Pareto fronts to which the solution belongs, and it is based on the dominance relationship among all solutions, whereas the crowding distance prefers the diversity among the solutions of the same Pareto front. Therefore, the solutions are ordered by rank: all the solutions from the first Pareto front appear first, then the solutions from the second Pareto front, and so on. Furthermore, within every rank (Pareto front), the solutions are ordered by crowding distance, appearing first the solutions with higher crowding distance.

The operations contained in the first "for" loop are repeated during a maximum number of cycles $cycles_{max}$, which is the considered stopping criterion. These operations performed in each cycle make the population evolve. Firstly, the population is divided into a number of $memes_{num}$ memeplexes ($Memeplexes$ contains $memes_{num}$ memeplexes). The distribution process is done by shuffling the ordered solutions, as it has been explained in Section 4.1. Secondly, the best global solution, X_{bestG} is selected, which will be used later in the improvement of the memeplexes.

Now, the operations included in the second and third "for" loops are performed for each memeplex $Memeplexes[m]$ (second loop) during a maximum number of improvements $improvs_{max}$ per memeplex (third loop). At the end of these two loops, the size of the population will be duplicated ($pop_{size} * 2$) because every original solution is stored before being replaced by its corresponding new generated solution. This is an improvement regarding the basic SFLA, where every original solution is directly replaced by its new solution generated. Therefore, in MOSFLA,

both the parent population and the offspring population are stored, and combined for finally obtaining the parent population for the next cycle (in this way, reducing the population again to its original size, pop_{size}). The goal of this modification is to give an opportunity to those original solutions that do not improve on their new solution generated (and would be discarded), but can improve on other new generated solutions of the population. Thereby, the algorithm does not lose any good solution known.

As said, this third loop performs the local search within each memeplex. A detailed explanation of this local search per memeplex is as follows. At the beginning, the best and worst local solutions, X_{bestL} and X_{worstL} respectively, from the memeplex m are selected. After that, the worst local solution is stored, before it is replaced.

Next, the mutation operator is executed (with a mutation probability p_m) in order to improve the worst local solution. This operator is explained in detail in Section 4.4. The first step is to mutate the best local solution X_{bestL}, and if the new solution X_{new} dominates the worst local solution, then X_{new} replaces it. If the new solution does not dominate the worst one, the second attempt is to mutate the best global solution X_{bestG}, and if the new solution X_{new} dominates the worst local solution, then X_{new} replaces it. Otherwise, a new solution is randomly generated and this will replace the worst local solution.

Finally, the memeplex $Memeplexes[m]$ is sorted by the dominance relationship. As, at the beginning of the local search within the memeplex, the memeplex is ordered, an ordered insertion of the new solution is the best way to keep the memeplex correctly ordered.

Once all the memeplexes have finished their improvements (local searches), the $memes_{num}$ memeplexes are combined into the population. After that, the values of the objective functions are calculated again and the population is ordered according to the rank and crowding distance. In this way, only the best half of the population (its original size, pop_{size}) will be used in the next cycle. At the end of the cycle, the non-dominated solutions are stored in the file NDS_file. It is possible that some solutions may not satisfy the length constraint indicated in Eq. (6). For this reason, the repair operation (described in Section 4.5) is performed on every solution before being stored in the file.

These detailed explanations of Algorithm 2 reveal that MOSFLA contains several improvements regarding SFLA. These contributions are the following ones. The development of a multi-objective optimization approach for the basic SFLA, which in turn includes the use and management of the non-dominated solution set, the rank and crowding operators, and the idea of ordering the memeplexes based on the dominance relationship of their solutions. Besides, the mutation and repair operators have been designed and developed specifically in a problem-aware way, i.e., both operators perform their operation by taking into account one of the main purposes of the query-oriented text summarization problem: the relevance of the sentences with the query. These operators are described in the following subsections.

4.4. Mutation

A mutation operator has been specifically designed and implemented for this problem and integrated into the MOSFLA approach. This operation consists of adding, removing, or exchanging a sentence from the summary based on the mutation selected. These three alternatives have the same probability of being selected and only one of them (randomly selected) will be performed in every mutation. Therefore, as only a sentence is mutated, the mutation probability $p_m = 1/n$, being n the number of sentences. It is important to highlight that the mutation is always performed, even when the affected sentence does not produce an improvement in the solution. The way in which the possible mutations are performed is:

- Adding a sentence to the summary. This action makes that a sentence from the document collection that is not contained in the summary will be included. The new sentence should improve the quality of the summary S. This means that the cosine similarity of a sentence $s_i \notin S$ with the query vector Q should be greater than the average of the cosine similarity of every sentence with the query vector. The condition in Eq. (8) sums up this explanation:

$$cosim(s_i, Q) > \frac{1}{n}\sum_{j=1}^{n} cosim(s_j, Q). \tag{8}$$

  The sentence $s_i \notin S$ is selected randomly from the document collection D, and if it fulfills the condition, it will be added to the summary. If it does not fulfill the condition, the next sentence $s_i \notin S$ is checked, and so on. If there is not any sentence meeting this condition, then the sentence $s_i \notin S$ with the greatest cosine similarity with the query vector will be added.
- Removing a sentence from the summary. This makes that a sentence from the summary will be discarded. The sentence to be deleted should not deteriorate the quality of the summary S. For this reason, the cosine similarity of a sentence $s_i \in S$ with the query vector Q should be lesser than the average of the cosine similarity of every sentence with the query vector, as indicated in the condition in Eq. (9):

$$cosim(s_i, Q) < \frac{1}{n}\sum_{j=1}^{n} cosim(s_j, Q). \tag{9}$$

  In the same way, the sentence $s_i \in S$ is selected from the summary S in a random way, and if it holds the condition, it is removed from the summary. If it does not hold the condition, the next sentence $s_i \in S$ is checked, and so on. If no sentence fulfills this condition, then the sentence $s_i \in S$ with the least cosine similarity with the query vector is removed.
- Exchanging a sentence from the summary with another from the document collection. This action makes that a sentence from the document collection that is not contained in the summary will replace another one in the summary. In this case, the mutation operation performed consists of removing a sentence from the summary and then adding a different sentence to the one removed.

4.5. Reparation

A repair operator has also been specifically designed, implemented, and integrated into the MOSFLA approach. This operation repairs those summaries that violate the length constraint defined in Eq. (6). The length of the summary is checked in both directions. If the summary has a length shorter than the length constraint, it is not repaired and it is discarded (because this happens very rarely), whereas if the summary has a length larger than the length constraint, it is repaired.

The repair operation is performed as follows. Let S^* be a summary that is longer than what is allowed. The reparation operation removes the sentences that have the least degrees of similarity with the query. This degree of similarity is calculated with the following score:

$$score_{s_i} = cosim(s_i, Q) + 10 \cdot \left(cosim(O^{S^*}, Q) - cosim(O^{S^*-s_i}, Q) \right), \tag{10}$$

where $cosim(O^{S^*}, Q)$ is the cosine similarity between the center of the summary O^{S^*} and the query vector Q, and $cosim(O^{S^*-s_i}, Q)$ is the cosine similarity between the center of the summary (excluding the sentence s_i) and the query vector Q. The second term of the score has an order more of magnitude because it measures the quality of the summary S^* when the sentence s_i is discarded. The center of the summary is a vector $O^{S^*} = (o_1, o_2, \ldots, o_m)$ whose components o_k are calculated as follows:

$$o_k = \frac{1}{n^{S^*}} \sum_{i=1}^{n} w_{ik} \cdot x_i, \quad k = 1, 2, \ldots, m, \tag{11}$$

Table 1
Characteristics of the TAC2009 datasets.

Description	Value
Number of topics	44
Number of documents (N) per topic	10
Average number of sentences (n) per topic	154
Average number of total terms per topic	5513
Average number of different terms (m) per topic	939

being n^{S^*} the number of sentences in the summary S^*.

The sentence with the lowest score is discarded. This repair operation is repeated until the length constraint is satisfied.

5. Experimental results

5.1. Datasets

The datasets have been provided by *Text Analysis Conference* (TAC, 2019), which is an open benchmark for query-oriented summarization evaluation from the National Institute of Standards and Technology (NIST, USA). Particularly, TAC2009 (McNamee & Dang, 2009) has been used for the experiments for comparative purposes with other approaches. TAC2009 contains a total of 44 topics, and every topic contains 10 documents based on news. In addition, there are four model summaries (made by human experts from NIST) for each topic, limited to 100 words, which have to be used as references to evaluate the quality of the generated summaries. Table 1 shows average counts of the datasets: the number of topics, the number of documents (N), the average number of sentences (n), the average number of total terms, and the average number of different terms (m) in each topic.

5.2. Performance evaluation metrics

In order to evaluate the performance of the summaries, the *Recall-Oriented Understudy for Gisting Evaluation* (ROUGE) metric has been used (Lin, 2004). ROUGE is the most commonly used measure in this type of summarization. It measures the quality of a computer-generated summary by counting the number of overlapping units between it and the reference summary (human-generated).

The ROUGE scores used for the evaluation have been ROUGE-1, ROUGE-2, and ROUGE-SU4, since they have been the ones used in the scientific literature to evaluate query-oriented summaries. ROUGE-N is the N-gram recall between the candidate summary and a set of reference summaries: $N = 1$ measures the amount of unigrams and $N = 2$ measures the amount of bigrams. ROUGE-N is calculated as:

$$ROUGE-N = \frac{\sum_{S \in RefSummaries} \sum_{N-gram \in S} Count_{match}(N-gram)}{\sum_{S \in RefSummaries} \sum_{N-gram \in S} Count(N-gram)}, \quad (12)$$

where $RefSummaries$ is the set of reference summaries, $Count_{match}(N-gram)$ is the number of N-grams co-occurring between the candidate summary and $RefSummaries$, and $Count(N-gram)$ is the number of N-grams in the reference summary S. ROUGE-SU4 measures the amount of overlap of skip-bigrams with a maximum gap length of 4.

5.3. Parameter settings

As the parameters pop_{size} (population size), $cycles_{max}$ (number of cycles/generations), and p_m (mutation probability) are general, they were established to $pop_{size} = 64$, $cycles_{max} = 1000$, and $p_m = 1/n$, being n the number of sentences in each case.

The specific parameters for MOSFLA are the number of memeplexes, $memes_{num}$, and number of improvements per memeplex, $improvs_{max}$. Therefore, these parameters were included in a parametric study. The relationship between these parameters is $memes_{num} \cdot improvs_{max} = pop_{size}$, which is 64. Furthermore, the values of $memes_{num}$ and $improvs_{max}$ cannot be equal to 1 (there cannot be a single memeplex, nor a single improvement can be performed on each memeplex). In conclusion, the settings of the tests experimented with these parameters are shown in Table 2.

Table 2
Settings of the tests experimented for the MOSFLA parameters. The values of the best configuration are shown in bold.

Test	$memes_{num}$	$improvs_{max}$
Test 1	2	32
Test 2	**4**	**16**
Test 3	8	8
Test 4	16	4
Test 5	32	2

The results obtained in this parametric study have reported that the setting with the values of $memes_{num} = 4$ and $improvs_{max} = 16$ has achieved the best average ROUGE scores, so they have been established as the configuration of these parameters.

The experimental results shown in this paper are the outcome from 31 independent runs (repetitions) performed for each experiment in order to provide reliable statistics. Experiments have been performed in a compute node with 4 processors AMD Opteron Abu Dhabi 6376 with 96 GB RAM. The algorithm has been implemented in C/C++ language, and it has been developed in Eclipse Platform on Ubuntu 18.04 LTS.

5.4. Selecting a single solution from the Pareto front

This subsection presents the methods considered for selecting a single solution from the Pareto front. The result obtained by a multi-objective optimization algorithm is not a single solution, but a set of non-dominated solutions. Any solution from this set is suitable to be selected as final solution. Nevertheless, it is necessary to follow some criteria to choose it.

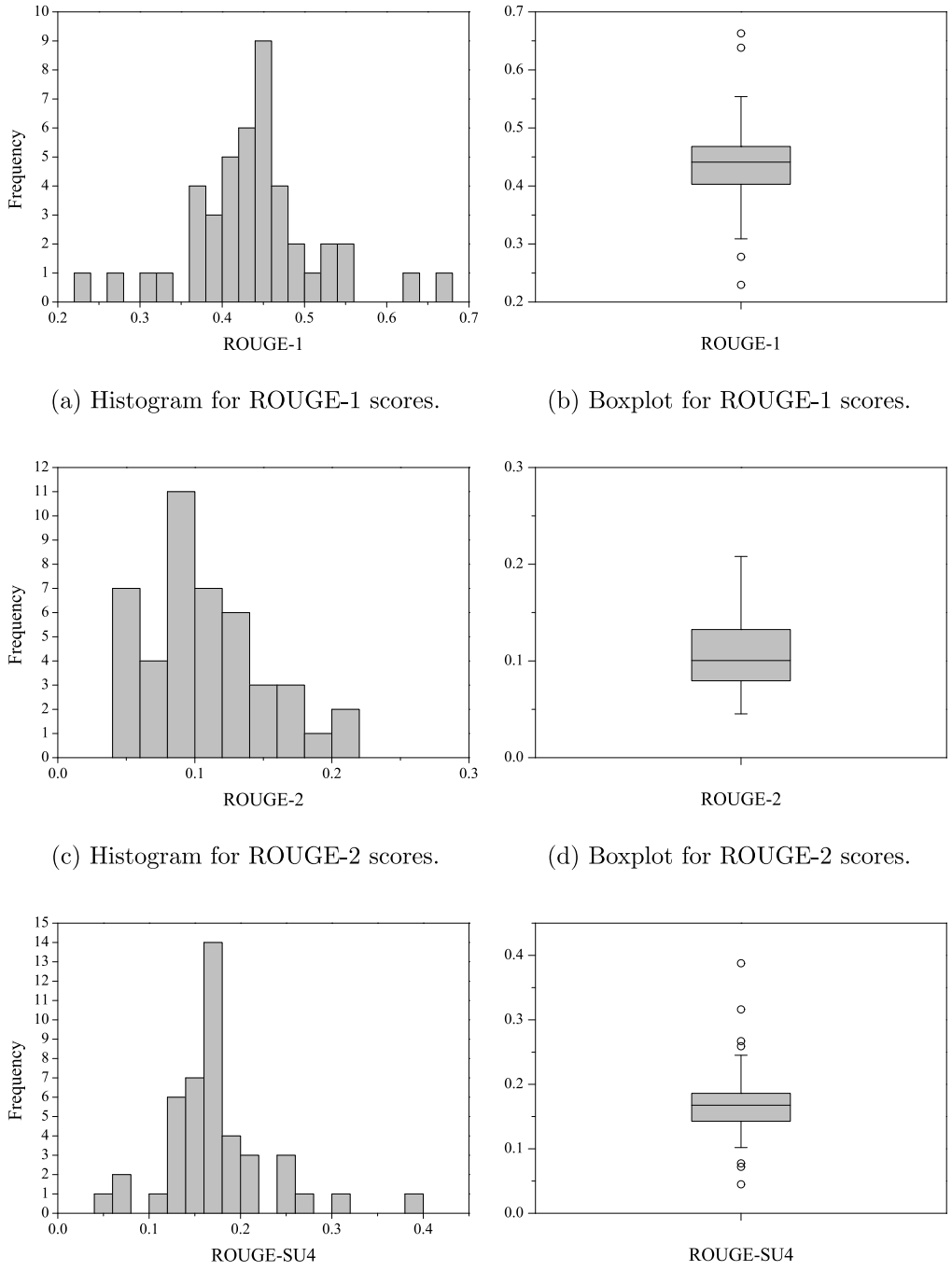
In the scientific literature, there are several methods for reducing the Pareto front to a single solution. Sanchez-Gomez et al. (2019) studied some methods based on the hypervolume, the consensus solution, the shortest distance to the ideal point (based on four distances: Euclidean, Manhattan, Chebyshev, and Mahalanobis), and the shortest distance to all points (based on five distances, the same previous four and Levenshtein); comparing and evaluating a total of eleven methods. This comparative study was applied to the generic extractive multi-document text summarization problem, so a similar study has been carried out in this work. The same eleven methods have been analyzed, evaluated, and compared (the detailed explanations of all the methods can be found in Sanchez-Gomez et al., 2019). The results have shown that the method of the shortest distance to the ideal point with the Mahalanobis distance has obtained the best average results in the three ROUGE scores. The second method has been the consensus solution, and the third one the method of the shortest distance to all points with Chebyshev distance. Therefore, in this work, the method used for selecting a single solution from the Pareto front has been the shortest distance to the ideal point with the Mahalanobis distance.

5.5. Results with the proposed approach

In this subsection, the results obtained by using the MOSFLA approach are presented and analyzed. A statistical analysis has been carried out with the results obtained for ROUGE-1, ROUGE-2, and ROUGE-SU4 scores for the 44 topics.

Table 3 includes the mean value, the median, the standard deviation, the first and third quartiles (Q_1 and Q_3), and the minimum and maximum values for the ROUGE scores based on the 31 repetitions (independent runs) per topic. The ROUGE score shown is the recall.

The results presented in Table 3 show the mean ROUGE scores obtained by MOSFLA: 0.440, 0.108, and 0.173 for ROUGE-1, ROUGE-2, and ROUGE-SU4 respectively. These average scores will be used for



(a) Histogram for ROUGE-1 scores. (b) Boxplot for ROUGE-1 scores.

(c) Histogram for ROUGE-2 scores. (d) Boxplot for ROUGE-2 scores.

(e) Histogram for ROUGE-SU4 scores. (f) Boxplot for ROUGE-SU4 scores.

Fig. 1. Histograms and boxplots obtained by MOSFLA for ROUGE-1, ROUGE-2, and ROUGE-SU4 scores for the 44 topics.

comparisons in the following subsection. Fig. 1 includes the histograms and boxplots for ROUGE-1, ROUGE-2, and ROUGE-SU4 scores.

Figs. 1(a), 1(c), and 1(e) show the histograms for ROUGE-1, ROUGE-2, and ROUGE-SU4 scores. These histograms represent the distribution of the ROUGE scores obtained for the 44 topics. Figs. 1(b), 1(d), and 1(f) present the boxplots for the ROUGE scores. They depict graphically the median (central segment of the box), Q_1 (lower segment), Q_3 (upper segment), minimum (lower whisker or outlier), and maximum (upper whisker or outlier). As it can be seen, there are few outliers per ROUGE score (4 in ROUGE-1, none in ROUGE-2, and 7 in ROUGE-SU4 out of 44 topics).

5.6. Comparison with results from other approaches

This subsection presents the results obtained by other approaches, which are compared to the ones provided with the proposed approach. Firstly, Table 4 represents the comparative results for the 44 topics. The results shown for MOSFLA are the mean value of ROUGE-1, ROUGE-2, and ROUGE-SU4 scores presented in Table 3. Only the average values of ROUGE scores are shown due to the fact that the other authors do not show other statistical measures, and the ROUGE score shown is the recall. Table 4 also includes (in brackets) the percentage improvement obtained by MOSFLA for every approach. The last row in the table presents the average results for the other approaches. The symbol "-" is

Table 3
Results obtained by MOSFLA for ROUGE-1, ROUGE-2, and ROUGE-SU4 scores for the 44 topics.

MOSFLA	ROUGE-1	ROUGE-2	ROUGE-SU4
Mean	0.440	0.108	0.173
Median	0.441	0.100	0.168
Standard deviation	0.080	0.042	0.060
Q_1	0.401	0.078	0.142
Q_3	0.469	0.133	0.187
Minimum	0.230	0.045	0.045
Maximum	0.663	0.208	0.388

used when the result for that approach is not available. The approach in Li and Li (2014) described two models, W-LDA and S-LDA, and conducted experiments with them, so they both have been included for comparison purpose. In the same way, the approach in Fors-Isalguez et al. (2018) developed two methods, NSGA-II TF-ISF and NSGA-II WE, for which experiments were carried out. For this reason, both methods have been included in Table 4. Regarding approaches in Marujo et al. (2015) and Marujo et al. (2016), unfortunately it is not possible to make comparisons with them because they used TAC2009 with a summary length of 250 words and this is not correct because TAC2009 only offers reference summaries (made by human experts from NIST) limited to 100 words. Therefore, in this table, MOSFLA is compared with a total of 12 approaches from other authors. As can be seen, these 12 approaches include (i) several evolutionary algorithms (EA), both multi-objective (NSGA-II TF-ISF and NSGA-II WE) and single-objective (EA BiProb); (ii) different algorithms based on LDA (Latent Dirichlet Allocation, the most used topic model), such as HS-LDA, S-sLDA, W-LDA, and S-LDA; and (iii) algorithms typically used in the field of query-oriented summarization, like LexRank, MEAD, KL-divergence, Manifold, and SVM.

The results reported in Table 4 demonstrate that MOSFLA outperforms the average ROUGE scores for almost all the approaches. Specifically, the average percentage improvements obtained have been 25.41%, 7.13%, and 30.22% for ROUGE-1, ROUGE-2, and ROUGE-SU4 scores, respectively. MOSFLA has improved the ROUGE-1 scores of all the approaches between 12.71% and 66.31% and all the ROUGE-SU4 scores between 15.54% and 54.94%, while for the ROUGE-2 scores MOSFLA has improved to 8 out of 12 approaches. Therefore, it can be concluded that MOSFLA has provided better results than the ones obtained in the scientific literature.

5.7. Multi-objective evaluation

This subsection contains the multi-objective evaluation of MOSFLA. As the compared approaches do not report on multi-objective evaluation metrics and they do not provide their source codes, a version of the standard NSGA-II (Deb et al., 2002) has been developed and adapted to the query-oriented extractive multi-document text summarization problem in order to compare with MOSFLA approach. To make fair comparisons, the parameters used for the standard NSGA-II have been the same as in MOSFLA (see Section 5.3).

The studied multi-objective evaluation metrics have been: Hypervolume (HV) and Inverted Generational Distance (IGD). HV is one of the most widely used performance indicators in the evolutionary multi-objective optimization field (Shang et al., 2021). For a set of non-dominated solutions or Pareto front, it measures the part of objective space that is dominated by them. Specifically, as this problem has two objective functions, this part is based on the area covered by the non-dominated solutions. The larger the hypervolume, the better the Pareto front. Regarding IGD metric, it is another well-known metric very used for assessing the quality of a set of non-dominated solutions (or Pareto front) with respect to the optimal Pareto front (Bezerra et al., 2017). It measures the average distance between each solution from the optimal Pareto front and the evaluated Pareto front, so the shorter the distance, the closer the Pareto front is to the optimal one. That is, the lower the IGD, the better the Pareto front.

Table 5 show the median values of the 31 repetitions for HV and IGD metrics obtained by MOSFLA and NSGA-II for each one of the 44 topics from TAC2009 datasets. In addition, the quartile deviation is also presented. It is defined as $(Q_3 - Q_1)/2$, and a representation of $Median \pm Quartile\_deviation$ is used in Table 5.

As can be appreciated in Table 5, the values obtained by MOSFLA for the HV indicator are better than the ones obtained by NSGA-II in most of the topics (33 out of 44). In fact, the average HV indicator is 0.270 for MOSFLA and 0.138 for NSGA-II. That is, MOSFLA's HV almost doubles NSGA-II's HV. As for IGD metric, MOSFLA also provides better values than NSGA-II in most of the topics (32 out of 44). Besides, the average IGD value is 0.625 for MOSFLA and 0.749 for NSGA-II. Therefore, it can be concluded that MOSFLA achieves a higher quality performance than the standard NSGA-II in terms of multi-objective evaluation.

These great improvements obtained by MOSFLA with respect to the standard NSGA-II are based on the following aspects: 1) it conducts multiple simultaneous searches over different memeplexes (sets of solutions); 2) it generates new solutions by considering the information provided by the best local solution within the processed memeplex and the best global solution in the population, also addressing stagnation situations by re-initializing solutions; and 3) it uses shuffling techniques to achieve a global exchange of knowledge among memeplexes, allowing a balanced distribution of promising solutions for improved optimization purposes.

6. Application to medicine texts

In this section, MOSFLA has been applied to a dataset based on medicine texts in order to show its applicability. The used dataset has been the (Topically Diverse Query Focus Summarization (TD-QFS), 2016) dataset, introduced by Baumel et al. (2016). This dataset is an expansion of the Query Chain Focused Summarization (QCFS) dataset (Baumel et al., 2014), and it contains four document collections gathered by medical experts about four different diseases: Asthma, Lung Cancer, Alzheimer's Disease, and Obesity. These document collections have been obtained from reliable sources related to each disease, such as U.S. National Institutes of Health, U.S. National Library of Medicine, U.S. Environmental Protection Agency, Palo Alto Medical Foundation, WebMD Medical Corporation, Cleveland Clinic, and Mayo Clinic, among others. Moreover, each document collection includes a set with several queries about the corresponding disease. All these queries have been extracted from PubMed query logs, the search engine from the U.S. National Library of Medicine, and they are concepts related to the corresponding disease as the causes, symptoms, diagnosis, medication, treatment, or other related diseases.

In this paper, the application of MOSFLA in the TD-QFS dataset has consisted in the generation of different query-oriented summaries according to different given queries, in order to analyze their differences. Specifically, the Asthma document collection has been used for this study. This collection contains documents obtained from other sources such as Asthma and Allergy Foundation of America, Asthma New Zealand, or National Eczema Association, among others, in addition to the sources listed above. Table 6 presents some characteristics of the Asthma document collection.

For this study, the following queries have been used: "atopic dermatitis" and "asthma allergy". The reason for choosing these two different queries is that it is intended to show that, from the same document collection, MOSFLA is capable of generating a summary that provides the most relevant information for each query. The parameter settings used in this study have been the same as in Section 5.3. Regarding the method for selecting the single solution from the Pareto front, the same method, the shortest Mahalanobis distance to the ideal

Table 4
Comparison of MOSFLA with other approaches for ROUGE-1, ROUGE-2, and ROUGE-SU4 scores for the 44 topics. The best values are shown in bold. The average scores and the percentage improvements obtained by MOSFLA are shown for every approach.

Approach	ROUGE-1		ROUGE-2		ROUGE-SU4	
MOSFLA	**0.440**		0.108		**0.173**	
NSGA-II TF-ISF (Fors-Isalguez et al., 2018)	0.265	(+66.31%)	0.090	(+21.00%)	–	
NSGA-II WE (Fors-Isalguez et al., 2018)	0.286	(+53.97%)	0.094	(+14.98%)	–	
EA BiProb (Bossard & Rodrigues, 2017)	0.386	(+14.12%)	0.117	(−7.64%)	–	
HS-LDA (Haghighi & Vanderwende, 2009)	0.360	(+22.23%)	0.100	(+7.91%)	0.128	(+35.21%)
S-sLDA (Li & Li, 2013)	0.390	(+12.71%)	**0.122**	(−11.41%)	0.149	(+16.31%)
W-LDA (Li & Li, 2014)	0.389	(+13.06%)	0.119	(−9.11%)	0.148	(+16.78%)
S-LDA (Li & Li, 2014)	0.390	(+12.74%)	0.121	(−10.39%)	0.150	(+15.54%)
LexRank (Erkan & Radev, 2004)	0.362	(+21.66%)	0.085	(+27.61%)	0.125	(+38.57%)
MEAD (Radev et al., 2004)	0.360	(+22.17%)	0.100	(+8.23%)	0.129	(+34.48%)
KL-divergence (Lin et al., 2006)	0.347	(+26.85%)	0.082	(+32.13%)	0.112	(+54.94%)
Manifold (Wan et al., 2007)	0.371	(+18.48%)	0.101	(+6.85%)	0.134	(+28.97%)
SVM (Li et al., 2009)	0.365	(+20.56%)	0.103	(+5.39%)	0.132	(+31.22%)
Average others	0.356	(+25.41%)	0.103	(+7.13%)	0.134	(+30.22%)

Table 5
Comparison of HV and IGD metrics ($Median \pm Quartile\_deviation$) obtained by MOSFLA and NSGA-II for the 44 topics from TAC2009. The best values are shown in bold.

Topic	HV metric		IGD metric	
	MOSFLA	NSGA-II	MOSFLA	NSGA-II
D0901A	**0.398 ± 0.0046**	0.071 ± 0.0399	**0.436 ± 0.0409**	0.469 ± 0.0505
D0902A	0.062 ± 0.0102	**0.075 ± 0.0012**	**0.843 ± 0.0728**	0.867 ± 0.0607
D0903A	**0.313 ± 0.0018**	0.023 ± 0.0137	**0.265 ± 0.0360**	0.416 ± 0.0461
D0904A	0.062 ± 0.0371	**0.327 ± 0.0049**	0.562 ± 0.0140	**0.467 ± 0.0005**
D0905A	0.609 ± 0.0769	**0.679 ± 0.1076**	**0.513 ± 0.0332**	0.696 ± 0.0562
D0906B	0.489 ± 0.0545	**0.547 ± 0.0411**	0.697 ± 0.0833	**0.578 ± 0.0884**
D0907B	**0.416 ± 0.0015**	0.045 ± 0.0036	**0.332 ± 0.0750**	0.498 ± 0.0172
D0908B	**0.086 ± 0.0009**	0.082 ± 0.0050	0.493 ± 0.0563	**0.402 ± 0.0291**
D0909B	**0.324 ± 0.0061**	0.025 ± 0.0016	**0.398 ± 0.0781**	0.527 ± 0.1152
D0910B	0.022 ± 0.0008	**0.024 ± 0.0018**	**0.429 ± 0.0454**	0.502 ± 0.0298
D0911C	**0.105 ± 0.0169**	0.002 ± 0.0075	**0.854 ± 0.0417**	1.157 ± 0.1121
D0912C	**0.377 ± 0.0050**	0.124 ± 0.0277	**0.875 ± 0.0114**	0.889 ± 0.0284
D0913C	0.117 ± 0.0328	**0.234 ± 0.0352**	**0.725 ± 0.0493**	0.807 ± 0.0736
D0914C	**0.242 ± 0.0378**	0.109 ± 0.0101	**0.311 ± 0.0558**	0.384 ± 0.0332
D0915C	**0.470 ± 0.0837**	0.012 ± 0.0005	**0.593 ± 0.0043**	0.681 ± 0.0687
D0916C	**0.541 ± 0.0842**	0.117 ± 0.0449	**0.789 ± 0.1061**	0.881 ± 0.1244
D0917C	0.006 ± 0.0003	**0.087 ± 0.0656**	0.984 ± 0.0897	**0.863 ± 0.0921**
D0918D	0.188 ± 0.0943	**0.590 ± 0.0961**	**0.672 ± 0.0721**	0.687 ± 0.1064
D0919D	**0.349 ± 0.0389**	0.322 ± 0.0015	0.672 ± 0.1031	**0.547 ± 0.0779**
D0920D	**0.184 ± 0.0006**	0.038 ± 0.0009	**0.603 ± 0.0323**	1.117 ± 0.0864
D0921D	**0.048 ± 0.0014**	0.036 ± 0.0038	**0.420 ± 0.0250**	0.443 ± 0.0172
D0922D	**0.441 ± 0.0887**	0.021 ± 0.0003	**0.270 ± 0.0534**	0.638 ± 0.0322
D0923D	**0.265 ± 0.0666**	0.011 ± 0.0006	**0.657 ± 0.0160**	0.721 ± 0.0283
D0924D	**0.407 ± 0.0646**	0.034 ± 0.0002	**0.237 ± 0.0056**	0.406 ± 0.0309
D0925E	**0.378 ± 0.0029**	0.017 ± 0.0014	**0.490 ± 0.0637**	0.812 ± 0.0981
D0926E	**0.296 ± 0.0251**	0.058 ± 0.0010	**1.171 ± 0.1087**	2.395 ± 0.1652
D0927E	0.261 ± 0.0652	**0.377 ± 0.0955**	0.675 ± 0.0996	**0.575 ± 0.0625**
D0928E	**0.308 ± 0.0195**	0.017 ± 0.0025	**0.591 ± 0.0132**	0.886 ± 0.0608
D0929E	**0.064 ± 0.0036**	0.063 ± 0.0054	0.277 ± 0.0113	**0.272 ± 0.0206**
D0930F	**0.265 ± 0.0101**	0.121 ± 0.0039	**0.855 ± 0.1354**	1.138 ± 0.1203
D0931F	0.391 ± 0.0171	**0.547 ± 0.0137**	0.682 ± 0.0666	**0.543 ± 0.0480**
D0932F	**0.145 ± 0.0636**	0.088 ± 0.0080	**0.873 ± 0.1336**	1.310 ± 0.1660
D0933F	**0.240 ± 0.0170**	0.094 ± 0.0039	**0.747 ± 0.0348**	0.953 ± 0.0870
D0934G	**0.404 ± 0.0009**	0.028 ± 0.0002	**0.271 ± 0.0725**	0.415 ± 0.0621
D0935G	**0.338 ± 0.0093**	0.122 ± 0.0034	**0.357 ± 0.0257**	0.633 ± 0.0313
D0936G	**0.041 ± 0.0550**	0.005 ± 0.0002	**0.917 ± 0.0741**	1.253 ± 0.1455
D0937G	**0.431 ± 0.0892**	0.315 ± 0.0894	1.071 ± 0.0882	**0.856 ± 0.0726**
D0938G	**0.345 ± 0.0294**	0.300 ± 0.0036	**0.364 ± 0.0078**	0.449 ± 0.0154
D0939H	**0.203 ± 0.0639**	0.053 ± 0.0003	**1.089 ± 0.0317**	1.278 ± 0.0757
D0940H	**0.495 ± 0.0347**	0.019 ± 0.0003	**0.693 ± 0.0128**	0.785 ± 0.0254
D0941H	**0.354 ± 0.0854**	0.059 ± 0.0124	0.308 ± 0.0551	**0.292 ± 0.0633**
D0942H	0.012 ± 0.0003	**0.015 ± 0.0003**	0.429 ± 0.0322	**0.352 ± 0.0458**
D0943H	**0.369 ± 0.0524**	0.114 ± 0.0027	**0.955 ± 0.0860**	1.055 ± 0.1055
D0944H	**0.020 ± 0.0001**	0.020 ± 0.0009	1.078 ± 0.0005	**1.078 ± 0.0004**
Average	**0.270**	0.138	**0.625**	0.749

In atopic dermatitis, the skin becomes extremely itchy. Scratching leads to redness, swelling, cracking, weeping clear fluid, and finally, crusting and scaling.

Most cases of atopic dermatitis can be diagnosed from a medical history and a physical exam.

Atopic dermatitis and hereditary eczema are interchangeable terms for an inflammatory condition of the skin characterized by erythema, pruritus, scaling, lichenification, and papulovesicles.

Antihistamine creams should not be used on atopic dermatitis rashes because they contain chemicals that can actually worsen the rash.

Allergic contact dermatitis occurs more commonly in adults.

Atopic dermatitis is often referred to as eczema, which is a general term for the several types of inflammation of the skin.

Asthma is a chronic lung disease that inflames and narrows the airways.

Asthma is a two-process condition (inflammation and bronchoconstriction) so to help alleviate both, a preventer and long acting bronchodilator are needed.

Allergies are grouped by the kind of trigger, time of year or where symptoms appear on the body: indoor and outdoor allergies, food allergies, latex allergies, insect allergies, skin allergies and eye allergies.

Allergic asthma is characterized by symptoms that are triggered by an allergic reaction. Exposure to dust mites can cause asthma in children who have not previously exhibited asthma symptoms.

Allergic asthma is airway obstruction and inflammation that is partially reversible with medication.

Fig. 2. Summaries generated for the queries "atopic dermatitis" (left) and "asthma allergy" (right) from the Asthma document collection of the TD-QFS dataset.

Table 6
Characteristics of the Asthma document collection from the TD-QFS dataset.

Description	Value
Number of sources	18
Number of documents	125
Number of sentences	1,924
Number of total terms	19,662
Number of different terms	2,284

point, has also been used (see Section 5.4). The summaries generated for each query are shown in Fig. 2.

As it can be appreciated in Fig. 2, the summary for the "atopic dermatitis" query is very different from the summary for the "asthma allergy" query, although they have been obtained from the same document collection. Both summaries contain sentences that are relevant for the corresponding queries, and these sentences are not redundant among them. Therefore, MOSFLA produces adequate query-oriented summaries, which can be customized according to the given query.

7. Conclusions

The query-oriented extractive multi-document text summarization has the peculiarity that implies the generation of a summary from a query given by the user. The query relevance and the redundancy reduction are considered as criteria to be optimized in this type of summaries. For this reason, a multi-objective optimization approach has been proposed.

In this paper, a memetic algorithm, Multi-Objective Shuffled Frog-Leaping Algorithm (MOSFLA), has been designed, implemented, and developed, for the first time, to solve this problem. MOSFLA is a population-based swarm intelligence algorithm, which includes new operators (e.g. mutation and repair) specifically designed for this problem and has been adapted for multi-objective optimization. In MOSFLA, the exploitation of the best solutions (local search) is performed in memeplexes (groups of solutions). Furthermore, the solutions are shuffled and then reorganized into new memeplexes regularly in order to ensure the global search. Moreover, some solutions are randomly generated and replace others in the population to give the chance to produce new explorations. After the statistical analysis of the results for 44 datasets, it can be concluded that MOSFLA provides better results than the ones of other approaches in the scientific literature (a total of 12 approaches from other authors have been used in the comparisons). MOSFLA has achieved an average percentage improvement of 25.41% in ROUGE-1 score, 7.13% in ROUGE-2 score, and 30.22% in ROUGE-SU4 score. Finally, the approach has been applied to medicine texts from the TD-QFS dataset as a case study, showing the goodness of the proposed approach with a real-world application.

In a future research, MOSFLA will be implemented in NeuroK software.[2] NeuroK is a collaborative e-learning platform based on neurodidactics and social networks principles (Calle-Alonso et al., 2017). The textual contents that students write in the different learning units or learning activities contained in the platform will be summarized. In this way, teachers can follow more easily the learning progress of the students. Moreover, they could evaluate the summaries of the students according to a query provided by the teacher.

CRediT authorship contribution statement

Jesus M. Sanchez-Gomez: Software, Validation, Formal analysis, Investigation, Data curation, Writing – original draft, Visualization. **Miguel A. Vega-Rodríguez:** Conceptualization, Methodology, Formal analysis, Resources, Writing – review & editing, Supervision, Project administration, Funding acquisition. **Carlos J. Pérez:** Conceptualization, Methodology, Formal analysis, Resources, Writing – review & editing, Supervision, Project administration, Funding acquisition.

Declaration of competing interest

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

[2] https://neurok.es/en/

Acknowledgments

This research has been supported by Ministry of Science, Innovation, and Universities - Spain and State Research Agency - Spain (Projects PID2019-107299GB-I00/AEI/10.13039/501100011033 and MTM2017-86875-C3-2-R), Junta de Extremadura - Spain (Projects GR18090 and GR18108), and European Union (European Regional Development Fund). Jesus M. Sanchez-Gomez is supported by Junta de Extremadura, Spain and European Union (European Social Fund) under the doctoral fellowship PD18057.

References

Alguliev, R. M., Aliguliyev, R. M., Hajirahimova, M. S., & Mehdiyev, C. A. (2011). MCMR: Maximum coverage and minimum redundant text summarization model. *Expert Systems with Applications*, *38*(12), 14514–14522. http://dx.doi.org/10.1016/j.eswa.2011.05.033.

Alguliev, R. M., Aliguliyev, R. M., & Mehdiyev, C. A. (2011). Sentence selection for generic document summarization using an adaptive differential evolution algorithm. *Swarm and Evolutionary Computation*, *1*(4), 213–222. http://dx.doi.org/10.1016/j.swevo.2011.06.006.

Baumel, T., Cohen, R., & Elhadad, M. (2014). Query-chain focused summarization. In *Proceedings of the 52nd annual meeting of the association for computational linguistics (Volume 1: long papers)* (pp. 913–922). Association for Computational Linguistics.

Baumel, T., Cohen, R., & Elhadad, M. (2016). Topic concentration in query focused summarization datasets. In *Proceedings of the thirtieth AAAI conference on artificial intelligence* (pp. 2573–2579). Association for the Advancement of Artificial Intelligence.

Bekele, E. G., & Nicklow, J. W. (2007). Multi-objective automatic calibration of SWAT using NSGA-II. *Journal of Hydrology*, *341*(3-4), 165–176. http://dx.doi.org/10.1016/j.jhydrol.2007.05.014.

Bezerra, L. C. T., López-Ibánez, M., & Stützle, T. (2017). An empirical assessment of the properties of inverted generational distance on multi- and many-objective optimization. In *International conference on evolutionary multi-criterion optimization* (pp. 31–45). Springer, http://dx.doi.org/10.1007/978-3-319-54157-0_3.

Bossard, A., & Rodrigues, C. (2017). An evolutionary algorithm for automatic summarization. In *Proceedings of the international conference on recent advances in natural language processing* (pp. 111–120). http://dx.doi.org/10.26615/978-954-452-049-6_017.

Calle-Alonso, F., Cuenca-Guevara, A., de la Mata Lara, D., Sanchez-Gomez, J. M., Vega-Rodríguez, M. A., & Perez Sanchez, C. J. (2017). Neurok: a collaborative e-learning platform based on pedagogical principles from neuroscience. In *Proceedings of the 9th international conference on computer supported education (Vol. 1)* (pp. 550–555). Science and Technology Publications.

Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, *6*(2), 182–197. http://dx.doi.org/10.1109/4235.996017.

Elbeltagi, E., Hegazy, T., & Grierson, D. (2007). A modified shuffled frog-leaping optimization algorithm: applications to project management. *Structure and Infrastructure Engineering*, *3*(1), 53–60. http://dx.doi.org/10.1080/15732470500254535.

Erkan, G., & Radev, D. R. (2004). Lexrank: graph-based lexical centrality as salience in text summarization. *Journal of Artificial Intelligence Research*, *22*, 457–479. http://dx.doi.org/10.1613/jair.1523.

Eusuff, M. M., & Lansey, K. E. (2003). Optimization of water distribution network design using the shuffled frog leaping algorithm. *Journal of Water Resources Planning and Management*, *129*(3), 210–225. http://dx.doi.org/10.1061/(ASCE)0733-9496(2003)129:3(210).

Eusuff, M. M., Lansey, K. E., & Pasha, F. (2006). Shuffled frog-leaping algorithm: a memetic meta-heuristic for discrete optimization. *Engineering Optimization*, *38*(2), 129–154. http://dx.doi.org/10.1080/03052150500384759.

Fan, W., & Bifet, A. (2013). Mining big data: current status, and forecast to the future. *ACM SIGKDD Explorations Newsletter*, *14*(2), 1–5. http://dx.doi.org/10.1145/2481244.2481246.

Fang, C., & Wang, L. (2012). An effective shuffled frog-leaping algorithm for resource-constrained project scheduling problem. *Computers & Operations Research*, *39*(5), 890–901. http://dx.doi.org/10.1016/j.cor.2011.07.010.

Fors-Isalguez, Y., Hermosillo-Valadez, J., & Montes-y-Gómez, M. (2018). Query-oriented text summarization based on multiobjective evolutionary algorithms and word embeddings. *Journal of Intelligent & Fuzzy Systems*, *34*(5), 3235–3244. http://dx.doi.org/10.3233/JIFS-169506.

Haghighi, A., & Vanderwende, L. (2009). Exploring content models for multi-document summarization. In *Human language technologies: Proceedings of the annual conference of the North American chapter of the association for computational linguistics* (pp. 362–370). Association for Computational Linguistics.

Hashimi, H., Hafez, A., & Mathkour, H. (2015). Selection criteria for text mining approaches. *Computers in Human Behavior*, *51*, 729–733. http://dx.doi.org/10.1016/j.chb.2014.10.062.

Huang, L., He, Y., Wei, F., & Li, W. (2010). Modeling document summarization as multi-objective optimization. In *Proceedings of the third international symposium on intelligent information technology and security informatics* (pp. 382–386). IEEE, http://dx.doi.org/10.1109/IITSI.2010.80.

Li, J. (2020). ROUGE Metric. URL: https://pypi.org/project/rouge-metric/. (Last accessed: 24-November-2021).

Li, J., & Li, S. (2013). A novel feature-based bayesian model for query focused multi-document summarization. *Transactions of the Association for Computational Linguistics*, *1*, 89–98. http://dx.doi.org/10.1162/tacl_a_00212.

Li, Y., & Li, S. (2014). Query-focused multi-document summarization: Combining a topic model with graph-based semi-supervised learning. In *Proceedings of the 25th international conference on computational linguistics (COLING): Technical papers* (pp. 1197–1207).

Li, L., Zhou, K., Xue, G.-R., Zha, H., & Yu, Y. (2009). Enhancing diversity, coverage and balance for summarization through structure learning. In *Proceedings of the 18th international conference on world wide web* (pp. 71–80). ACM, http://dx.doi.org/10.1145/1526709.1526720.

Lin, C.-Y. (2004). ROUGE: A package for automatic evaluation of summaries. In *Text summarization branches out: Proceedings of the ACL-04 workshop (Vol. 8)* (pp. 74–81). Association for Computational Linguistics.

Lin, C.-Y., Cao, G., Gao, J., & Nie, J.-Y. (2006). An information-theoretic approach to automatic evaluation of summaries. In *Human language technology: Proceedings of the annual conference of the North American chapter of the association for computational linguistics* (pp. 463–470). Association for Computational Linguistics, http://dx.doi.org/10.3115/1220835.1220894.

Marujo, L., Ling, W., Ribeiro, R., Gershman, A., Carbonell, J., de Matos, D. M., & Neto, J. P. (2016). Exploring events and distributed representations of text in multi-document summarization. *Knowledge-Based Systems*, *94*, 33–42. http://dx.doi.org/10.1016/j.knosys.2015.11.005.

Marujo, L., Ribeiro, R., de Matos, D. M., Neto, J. P., Gershman, A., & Carbonell, J. (2015). Extending a single-document summarizer to multi-document: a hierarchical approach. In *Proceedings of the fourth joint conference on lexical and computational semantics* (pp. 176–181). Association for Computational Linguistics.

McNamee, P., & Dang, H. T. (2009). Overview of the TAC 2009 knowledge base population track. In *Text analysis conference (Vol. 17)* (pp. 111–113). National Institute of Standards and Technology (NIST).

Murugan, P., Kannan, S., & Baskar, S. (2009). NSGA-II algorithm for multi-objective generation expansion planning problem. *Electric Power Systems Research*, *79*(4), 622–628. http://dx.doi.org/10.1016/j.epsr.2008.09.011.

Porter, M. (2020). The porter stemming algorithm. http://www.tartarus.org/martin/PorterStemmer/. (Last accessed: 24-November-2021).

Radev, D. R., Jing, H., Styś, M., & Tam, D. (2004). Centroid-based summarization of multiple documents. *Information Processing & Management*, *40*(6), 919–938. http://dx.doi.org/10.1016/j.ipm.2003.10.006.

Ramesh, S., Kannan, S., & Baskar, S. (2012). Application of modified NSGA-II algorithm to multi-objective reactive power planning. *Applied Soft Computing*, *12*(2), 741–753. http://dx.doi.org/10.1016/j.asoc.2011.09.015.

Salton, G., & Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. *Information Processing & Management*, *24*(5), 513–523. http://dx.doi.org/10.1016/0306-4573(88)90021-0.

Sanchez-Gomez, J. M., Vega-Rodríguez, M. A., & Pérez, C. J. (2018). Extractive multi-document text summarization using a multi-objective artificial bee colony optimization approach. *Knowledge-Based Systems*, *159*, 1–8. http://dx.doi.org/10.1016/j.knosys.2017.11.029.

Sanchez-Gomez, J. M., Vega-Rodríguez, M. A., & Pérez, C. J. (2019). Comparison of automatic methods for reducing the pareto front to a single solution applied to multi-document text summarization. *Knowledge-Based Systems*, *174*, 123–136. http://dx.doi.org/10.1016/j.knosys.2019.03.002.

Shang, K., Ishibuchi, H., He, L., & Pang, L. M. (2021). A survey on the hypervolume indicator in evolutionary multi-objective optimization. *IEEE Transactions on Evolutionary Computation*, *25*(1), 1–20. http://dx.doi.org/10.1109/TEVC.2020.3013290.

TAC (2019). Text analysis conference. https://tac.nist.gov/. Last accessed: 24-November-2021.

Tang, J., Zhang, R., Wang, P., Zhao, Z., Fan, L., & Liu, X. (2020). A discrete shuffled frog-leaping algorithm to identify influential nodes for influence maximization in social networks. *Knowledge-Based Systems*, *187*(104833), 1–12. http://dx.doi.org/10.1016/j.knosys.2019.07.004.

Topically Diverse Query Focus Summarization (TD-QFS) (2016). Natural language processing lab. https://www.cs.bgu.ac.il/~talbau/TD-QFS/dataset.htmlLast. (Last accessed: 24-November-2021).

Wan, X. (2008). An exploration of document impact on graph-based multi-document summarization. In *Proceedings of the conference on empirical methods in natural language processing* (pp. 755–762). Association for Computational Linguistics.

Wan, X., Yang, J., & Xiao, J. (2007). Manifold-ranking based topic-focused multi-document summarization. In *Proceedings of the 20th international joint conference on artifical intelligence (Vol. 7)* (pp. 2903–2908).

Zajic, D. M., Dorr, B. J., & Lin, J. (2008). Single-document and multi-document summarization techniques for email threads using sentence compression. *Information Processing & Management*, *44*(4), 1600–1610. http://dx.doi.org/10.1016/j.ipm.2007.09.007.