Swarm and Evolutionary Computation 91 (2024) 101721

Contents lists available at ScienceDirect

Swarm and Evolutionary Computation

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

An indicator-based multi-objective variable neighborhood search approach for query-focused summarization

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, Campus Universitario s/n, 10003 Cáceres, Spain[1]

[b] Departamento de Matemáticas, Universidad de Extremadura, Campus Universitario s/n, 10003 Cáceres, Spain[1]

ARTICLE INFO

Keywords:
Evolutionary computation
Variable neighborhood search
Multi-objective optimization
Indicator-based optimization
Query-focused summarization

ABSTRACT

Currently, automatic multi-document summarization is an interesting subject in numerous fields of study. As a part of it, query-focused summarization is becoming increasingly important in recent times. These methods can automatically produce a summary based on a query given by the user, including the most relevant information from the query at the same time as the redundancy among sentences is reduced. This can be achieved by developing and applying a multi-objective optimization approach. In this paper, an Indicator-based Multi-Objective Variable Neighborhood Search (IMOVNS) algorithm has been designed, implemented, and tested for the query-focused extractive multi-document summarization problem. Experiments have been carried out with datasets from Text Analysis Conference (TAC). The results were evaluated using the Recall-Oriented Understudy for Gisting Evaluation (ROUGE) metrics. IMOVNS has greatly improved the results presented in the scientific literature, providing improvement percentages in ROUGE metric reaching up to 69.24% in ROUGE-1, up to 57.70% in ROUGE-2, and up to 77.37% in ROUGE-SU4 scores. Hence, the proposed IMOVNS offers a promising solution to the query-focused summarization problem, thus highlighting its efficacy and potential for enhancing automatic summarization techniques.

1. Introduction

Nowadays, the amount of digital documents contained in the World Wide Web is growing in an exponential way. This is mainly due to the vertiginous development of information and communication technologies. When a user tries to obtain information on a specific topic, it is difficult to handle such a vast volume of information. For this reason, the use of text mining tools becomes necessary, as they are capable of extracting specific textual information from a large document collection [1]. Furthermore, some approaches can automatically generate summaries from a document collection [2]. The generated summaries cover the most relevant information and avoid redundant information.

There are several methods to produce automatic summaries. Firstly, automatic text summarization may be generic or query-focused. Generic methods produce a summary without needing any user information [3]. On the other hand, query-focused methods require user information. This is usually provided as a query, that is a sentence of a specific topic for the user [4]. Thus, query-focused methods produce a summary according to the user information. Secondly, automatic summarization methods can be abstractive or extractive [5]. In abstractive methods, the summary is composed by words and sentences that may not exist in the original document collection, whereas in extractive methods the summary is only made up by sentences that exist in the original source. Another way of classifying automatic summarization methods is based on the number of documents: single-document methods summarize the information from only one document, whereas multi-document methods sum up information from a document collection [6]. Finally, automatic summarization methods can be: single-objective or multi-objective. Single-objective methods focus solely on optimizing one objective function, by assigning weights to the different criteria [7]. Conversely, multi-objective methods aim to optimize several objective functions simultaneously [8].

Among the existing metaheuristics, the Variable Neighborhood Search (VNS) algorithm emerged as a prominent alternative for addressing combinatorial optimization and other optimization problems [9]. For this reason, an Indicator-based Multi-Objective Variable Neighborhood Search (IMOVNS) algorithm has been designed, implemented, and tested in this paper for solving the query-focused extractive multi-document summarization problem. Two objective functions, based on the criteria of query relevance and redundancy reduction, have been

* 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.swevo.2024.101721
Received 26 July 2023; Received in revised form 15 July 2024; Accepted 26 August 2024
Available online 28 August 2024

Table 1
Summary of the related work.

Article	Approach	Model	Multi-objective	ROUGE scores
[12]	S-sLDA	Sentence feature-based Bayesian model	No	1, 2, SU-4
[13]	LexRank	Stochastic graph-based lexical centrality	No	1, 2, SU-4
[14]	MEAD	Cluster centroids with topic detection	No	1, 2, SU-4
[15]	Manifold	Manifold-ranking and greedy algorithm	No	1, 2, SU-4
[16]	SVM	Learning framework and cutting plane algorithm	No	1, 2, SU-4
[17]	W-LDA and S-LDA	Two-layer graph ranking algorithm	No	1, 2, SU-4
[18]	KL-divergence	Sentence selection strategy	No	1, 2, SU-4
[19]	Hiersum	Hierarchical LDA topic model	No	1, 2, SU-4
[20]	KP-Centrality	Single-layer and waterfall architectures	No	1, 2
[21]	Fuzzy fingerprint	Classification-based event detection	No	1, 2, SU-4
[22]	BiProb EABS	Evolutionary algorithm	No	1, 2
[23]	QOS-MOO-TFISF and QOS-MOO-WE	NSGA-II	Yes	1, 2
[24]	MOSFLA	Multi-objective memetic algorithm	Yes	1, 2, SU-4

formulated to be maximized. Experimentation has been performed with datasets from Text Analysis Conference (TAC) [10]. The Recall-Oriented Understudy for Gisting Evaluation (ROUGE) metrics [11] have been used to evaluate the summary quality. The main contributions of this paper are:

- The problem of query-focused multi-document summarization has been approached as a multi-objective optimization problem, where the objective functions are defined in terms of query relevance and redundancy reduction.
- For the first time, an indicator-based multi-objective optimization approach based on VNS algorithm has been implemented and applied for this problem.
- The new algorithm includes several advances regarding standard VNS (a restarting process, a list of tabu solutions, and a set with the best solutions found), along with a mutation operator and a repair operation specifically designed for this problem.
- The results have been compared to other competitive approaches proposed in the scientific literature, showing a great improvement in quality terms.

The remainder of this paper is as follows. Section 2 includes the related work. In Section 3, query-focused multi-document summarization is formulated as a multi-objective optimization problem. Section 4 presents the Indicator-based Multi-Objective Variable Neighborhood Search algorithm. In Section 5, evaluation metrics, datasets, parameter settings, experimental results, and comparisons with other approaches are detailed. Finally, Section 6 provides the conclusions.

2. Related work

This section systematically reviews a series of proposals that have tackled the query-focused multi-document summarization problem. The review is presented in chronological order, offering insights into the evolution of research in this domain over time.

In [12], a pioneering model called Supervised Latent Dirichlet Allocation (S-sLDA) was introduced. It integrated both topic models and feature-based supervised methods, which transformed the task of determining optimum feature weights into an optimization problem. In this proposal, several models applied for query-focused summarization were studied, evaluated, and compared: a graph-based LexRank method, which calculates the relative significance of text parts based on eigenvector centrality [13]; another graph-based method, MEAD, that generates summaries by identifying cluster centroids produced by tracking system and topic detection [14]; a Manifold-based approach, which ranks sentences based on manifold-ranking and uses a greedy algorithm for punishing those with large levels of overlay [15]; and a learning framework developed by using the Support Vector Machine (SVM) method [16].

A semi-supervised learning method was introduced in [17]. This approach augmented the conventional graph ranking algorithm operating across two layers, and it was based on a pair of variants of the LDA method: the Sentence-level LDA (S-LDA) and Word-level LDA (W-LDA). Additionally, two other models were also compared in this proposal: an information theoretic method developed by using the KL-divergence based strategy [18]; and Hiersum, a modification of the hierarchical LDA method [19].

Later, in [20], the KP-Centrality model was proposed, examining both single-layer and waterfall architectures for summarization. The single-layer architecture accumulates summaries connected by chronological order, whereas the waterfall architecture joins the intermediate ones. MEAD and LexRank methods were also compared in this work. A classification-based event detection technique called Fuzzy Fingerprint was presented in [21], which used bag-of-words and skip-ngram as text distributed representations. This proposal was also evaluated against MEAD and LexRank models. Furthermore, [22] proposed a sentence extraction method employing an evolutionary algorithm. This approach computed the bigram probability distribution (BiProb EABS) of tokens in the source documents along with the probability distribution from the summaries.

Finally, the presented problem was addressed from a multi-objective optimization point of view by using both the standard *tf-isf* representation (QOS-MOO-TFISF) and the word embedding representation (QOS-MOO-WE) [23]. Additionally, in [24], a Multi-Objective Shuffled Frog-Leaping Algorithm (MOSFLA) was proposed, with a focus on optimizing the objective functions of query relevance and redundancy reduction.

All the previous approaches used ROUGE metrics, particularly ROUGE-1, ROUGE-2, and ROUGE-SU4 scores for experimentation. Moreover, all of them used TAC2009 datasets. For this reason, the same metrics and datasets have been considered in this paper. Table 1 contains the approaches reviewed, including the model used, the multi-objective nature, and the evaluated ROUGE scores.

As a conclusion from the review of the related work, it has been observed that the vast majority of the papers reviewed did not use multi-objective optimization approaches to solve the presented problem. As the definition of the query-focused multi-document summarization problem presents objectives that are conflicting with each other, the use of multi-objective optimization approaches should provide better performance. Unfortunately, there are very few multi-objective optimization approaches in the literature applied to this particular problem. In fact, only two proposals have been found: [23], which used a very standard algorithm, NSGA-II, and [24], that was based on a very specific, memetic algorithm. Therefore, by using an indicator-based multi-objective optimization approach that adapts the VNS metaheuristic technique, it is intended to improve the existing results in the scientific literature.

3. Problem definition

This section formulates the query-focused summarization problem. A vector-based word model is utilized. It is the most common model to feature words from texts in this field. In this model, a vector of words represents a sentence, and the resemblance between pairs of sentences is calculated using cosine similarity.

3.1. Cosine measure

Let $T = \{t_1, t_2, \dots, t_m\}$ be the set containing the m distinct terms of the document collection D. The document collection contains a total of n sentences. Every sentence $s_i \in D$ is symbolized as a vector with m components as $s_i = (w_{i1}, w_{i2}, \dots, w_{im}), i = 1, 2, \dots, n$, where each component w_{ik} corresponds to the weight of the term t_k in the sentence s_i. This weight can be obtained by means of the *term-frequency inverse-sentence-frequency* (*tf-isf*) scheme [25] as follows:

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

The term tf_{ik} refers to the term frequency that counts how many times t_k appears in the sentence s_i, and the second term is related to the inverse sentence frequency that measures the number of sentences n_k in which the term t_k occurs.

Once the sentence representation has been set, the similarity measure is defined. The cosine similarity calculates the resemblance between the pair of sentences s_i and s_j by:

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

3.2. Formulation of the optimization problem

The document collection is represented as a set of N documents $D = \{d_1, d_2, \dots, d_N\}$ or as a set of n sentences from the collection, i.e., $D = \{s_1, s_2, \dots, s_n\}$. The goal of the optimization problem is to generate a summary $S \subset D$ that includes a number n_S of sentences from the document collection D. The two following aspects must be held:

- The generated summary has to include the most relevant sentences according to the query provided by the user (query relevance).
- The generated summary must contain sentences that are different from each other (redundancy reduction).

The joint consideration of these two aspects involves the simultaneous optimization of the query relevance and the redundancy reduction. Specifically, while query relevance tries to choose the sentences that are most closely to the user's query (those that have the most common words with the user's query), redundancy reduction tries to avoid that similar sentences are present in the summary (precisely those that have many words in common). In the extreme case, the query relevance would generate a summary with only the most similar sentences to the user's query and, therefore, with a high redundancy. In contrast, redundancy reduction would select only the most diverse sentences from the collection, which could have a low query relevance. So, by definition, the two criteria to optimize are conflicting with each other and, therefore, this optimization problem is prone to be addressed through a multi-objective optimization point of view.

Before defining the objective functions to be optimized, it is necessary to introduce the representation of a solution X. Let $x_i \in \{0, 1\}$ be the binary decision variable that considers the presence or absence of the sentence s_i in the generated summary S, i.e.:

$$x_i = \begin{cases} 1 & \text{if } s_i \in S \\ 0 & \text{if } s_i \notin S. \end{cases} \tag{3}$$

This leads to the solution vector $X = (x_1, x_2, \dots, x_n)$.

The first objective function concerns to the criterion of query relevance. It is formulated as the cosine similarity between every sentence from the summary, $s_i \in S$, and the query vector, $Q = (q_1, q_2, \dots, q_m)$. This query vector symbolizes the query given by the user in the same way as a sentence, so its weights q_k can be calculated as detailed in Eq. (1). Thus, the following objective function should be maximized:

$$\Phi_{QR}(X) = \sum_{i=1}^{n} cosim(s_i, Q) \cdot x_i. \tag{4}$$

The second objective function refers to the criterion of the redundancy reduction. It is attained by minimizing the cosine similarity between every pair of sentences in the summary $s_i, s_j \in S$. This is equivalent to maximize:

$$\Phi_{RR}(X) = -\sum_{i=1}^{n-1} \sum_{j=i+1}^{n} cosim(s_i, s_j) \cdot x_i x_j. \tag{5}$$

Finally, the summary length constraint (named L and measured in words) must be taken into account in the optimization process. Hence, the query-focused multi-document summarization problem can be addressed from a multi-objective optimization viewpoint by solving:

$$\max \ \Phi(X) = \{\Phi_{QR}(X), \Phi_{RR}(X)\}, \tag{6}$$

$$\text{subject to } L - \varepsilon \le \sum_{i=1}^{n} l_i \cdot x_i \le L + \varepsilon. \tag{7}$$

The summary length constraint includes a length tolerance ε which is calculated as:

$$\varepsilon = \max_{i=1,2,\dots,n} l_i - \min_{i=1,2,\dots,n} l_i, \tag{8}$$

being l_i the length in words of the sentence s_i.

4. Indicator-based multi-objective variable neighborhood search

This section describes the Indicator-based Multi-Objective Variable Neighborhood Search (IMOVNS) algorithm developed in this work. It is built based on the following subsections: the basic Variable Neighborhood Search (VNS) algorithm, the preprocessing steps, the main steps of the IMOVNS algorithm, and its main operators. The concepts and mathematical formulation of the indicator-based multi-objective search strategy are detailed in [26].

4.1. Basic variable neighborhood search algorithm

The VNS algorithm is a metaheuristic specifically designed to solve combinatorial optimization and global optimization problems [9]. It is based on systematic changes of neighborhood combined with a local search. In addition, this algorithm has been successfully applied in several real-life problems, such as nurse rostering [27], design of supplementary damping controllers for small-signal stability analysis [28], vehicle routing problem [29], and the coordinated serial-batching scheduling [30], among others.

VNS algorithm starts from a single initial solution, and it explores increasingly distant neighborhoods of the current solution. The exploration process is repeated during a maximum number of evaluations or a maximum execution time. In the event that an improvement is achieved, the current (best) solution is updated and the neighborhood exploration is restarted. Otherwise, if the solution does not improve after a maximum number of neighborhood changes or trials, then the process starts again until the stopping condition is met. The steps of the VNS algorithm are detailed in Algorithm 1.

In line 1, the single solution *sol* is initialized randomly. Then, the current number of evaluations $evals_{num}$ is initialized to 1, and the fitness value of this solution is calculated (lines 2–3). After that, the steps from lines 4 to 17 are repeated until the maximum number of evaluations

Algorithm 1 Pseudocode of VNS algorithm.

```
1: sol ← initRandomSolution()
2: evals_num ← 1
3: sol.fitness ← calculateFitness(sol)
4: while evals_num ≤ evals_max do
5:    neigh_num ← 1
6:    while neigh_num ≤ neigh_max do
7:       mSol ← mutateSolution(sol)
8:       evals_num ← evals_num + 1
9:       mSol.fitness ← calculateFitness(mSol)
10:      if mSol.fitness ≻ sol.fitness then
11:         sol ← mSol
12:         neigh_num ← 1
13:      else
14:         neigh_num ← neigh_num + 1
15:      end if
16:   end while
17: end while
```

$evals_{max}$ is reached. This is one possible stopping condition, and another one is the execution time.

In line 5, the current neighborhood $neigh_{num}$ is initialized to 1, and the steps from lines 6 to 16 are repeated until the maximum number of neighborhoods $neigh_{max}$ is reached. Now, the solution sol is mutated in order to improve it, and the result is stored in the solution $mSol$ (line 7). Then, the number of evaluations $evals_{num}$ is increased in line 8. In line 9, the fitness value of the mutated solution $mSol$ is calculated and then compared with the fitness value of the solution sol. If the fitness value of the mutated solution is better than the one of the current (best) solution (line 10), then the mutated solution replaces this solution (line 11). In this case, the current number of neighborhood is restored to 1 in line 12. Otherwise, the current number of neighborhood is increased (line 14).

4.2. Preprocessing steps

All the documents from collection D must be processed before the IMOVNS algorithm is applied. The following four steps are performed for every document:

1. Sentence segmentation. The sentences from each document are extracted separately, also defining their beginning and their ending.
2. Word tokenization. The words from each sentence are isolated with a determined token (blank space). In this step, punctuation marks, exclamation marks, interrogation marks, and others are eliminated.
3. Stop word removal. The words without meaning (articles, prepositions, conjunctions, and others) are removed. The list of stop words of ROUGE package [31] is used.
4. Word stemming. The suffix of the words is removed by using the Porter stemming algorithm, maintaining the root [32]. In this way, the words with the same lexical root can be computed as the same term.

4.3. Main steps of the IMOVNS algorithm

In addition to adapting VNS to indicator-based multi-objective optimization, the IMOVNS algorithm developed here includes a series of advances with respect to the basic VNS. Firstly, a restarting process of the current solution has been implemented to reset the solution when the maximum number of neighborhoods has been reached. This allows the algorithm to escape local optima and continue exploring the solution space in an effective way. Secondly, a list of tabu solutions is managed to store the solutions that have already been used, ensuring that they are not selected as restarting points again. This enhancement helps to diversify the search and avoid considering the same solutions repeatedly. Finally, the best solutions found during the search are stored in a set of saved solutions. These elite solutions serve as valuable reference points and contribute to maintaining the quality and diversity of the population, thereby improving the algorithm's performance. Algorithm 2 outlines the main steps of IMOVNS.

Algorithm 2 IMOVNS pseudocode.

```
1: SavedSols ← ∅
2: TabuList ← ∅
3: sols_num ← 0
4: evals_num ← 0
5: while evals_num ≤ evals_max do
6:    if evals_num = 0 then
7:       sol ← initRandomSolution()
8:       evals_num ← evals_num + 1
9:       sol.fitness ← 0
10:      SavedSols ← insertSolutionByFitness(SavedSols, sol)
11:      sols_num ← sols_num + 1
12:   else
13:      sol ← getBestSolutionByFitness(SavedSols, TabuList)
14:   end if
15:   TabuList ← insertSolution(TabuList, sol)
16:   neigh_num ← 1
17:   while neigh_num ≤ neigh_max and evals_num ≤ evals_max do
18:      mSol ← mutateSolution(sol)
19:      evals_num ← evals_num + 1
20:      mSol.fitness ← calculateFitness(SavedSols, mSol, sols_num)
21:      if mSol.fitness < sol.fitness then
22:         sol ← mSol
23:         neigh_num ← 1
24:         SavedSols ← insertSolutionByFitness(SavedSols, sol)
25:         sols_num ← sols_num + 1
26:      else
27:         neigh_num ← neigh_num + 1
28:      end if
29:   end while
30: end while
31: saveBestSolutions(SavedSols, sols_num)
```

Firstly, the set $SavedSols$ that will store the best solutions and the list $TabuList$ that will store the tabu solutions are initialized in lines 1 and 2. Then, in lines 3 and 4, the current number of saved solutions $sols_{num}$ and the current number of evaluations, $evals_{num}$, are initialized to 0. After that, the steps from lines 5 to 30 are repeated during a maximum number of evaluations $evals_{max}$.

The steps from lines 6 to 14 refer to the restarting process of the solution sol. First, the current number of evaluations is checked. If it is the first evaluation, then the solution is initialized randomly and the number of evaluations is increased (lines 7 and 8). Then, in line 9, the indicator-based fitness value of sol is initialized to 0 due to the fact that it is not possible to calculate it at this moment (there is only one solution). After that, in lines 10 and 11, the solution is stored in $SavedSols$ in ascending order according to the indicator-based fitness value, and the number of saved solutions $sols_{num}$ is increased. In the case that $evals_{num}$ is greater than 0 (line 12), then the best solution stored in $SavedSols$, which has not been previously used (it is not in the $TabuList$), is copied in sol (line 13). The best solution is the one with the best indicator-based fitness value (the smallest one), so the set $SavedSols$ is sorted in ascending order according to the fitness value. In this way, the solution copied from the set $SavedSols$ is located in the first positions (and not included in the $TabuList$).

Now, in line 15, the solution sol is stored in $TabuList$, so it cannot be selected again as restarting point. After that, the current number of neighborhood $neigh_{num}$ is initialized to 1 in line 16, and the steps from lines 17 to 29 are repeated until the maximum number of neighborhoods $neigh_{max}$ is reached. The current number of evaluations $evals_{num}$ is also checked to ensure that the maximum number of evaluations $evals_{max}$ is not exceeded. The solution sol is mutated in line 18 in order to improve it, and the new mutated solution is stored in $mSol$ (the mutation operator is explained in detail in the following subsection). Then, in line 19, the number of evaluations is increased. In line 20, the indicator-based fitness value of the mutated solution $mSol$ is calculated, and if it is less than the indicator-based fitness value of sol (line 21), then the mutated solution is better, and therefore, it replaces the current one (line 22) and the number of neighborhood $neigh_{num}$ is restarted to 1 (line 23). In lines 24 and 25, the new solution sol is stored in $SavedSols$ in ascending order according to the indicator-based fitness value, also increasing the number of saved solutions $sols_{num}$. On the other side, in the case that the mutated solution $mSol$ does not improve the current solution sol, then the number of neighborhood is increased (line 27).

Finally, at the end of the two loops, the set of best solutions $SavedSols$ is stored in line 31. In this step, each solution is checked to verify if the length constraint defined in Eq. (7) is met. If it is not fulfilled, then the repair operator is carried out, which is described in detail in the following subsection.

4.4. Operators in IMOVNS

Two important operators in IMOVNS algorithm are defined and explained in this subsection: the mutation operator and the repair operator.

Firstly, the mutation operation involves the addition, the removal, or the exchange of a single sentence in the summary depending on which mutation is chosen. These three types of mutations have the same probability of being chosen, and only one of them will be carried out when applying this operator. Hence, the mutation probability is $p_m = 1/n$, since only one sentence will always be mutated from the total of n sentences. The three available mutations are described below:

- Addition of a sentence: a sentence from the document collection which does not exist in the summary S is added. This sentence $s_i \notin S$ must outperform the current summary quality, so its cosine similarity with the query vector Q must be larger than the average cosine similarity, i.e.:

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

  The new sentence $s_i \notin S$ is chosen in a random way, and it will be added to the summary if it meets Eq. (9). Otherwise, another sentence $s_i \notin S$ will be tested until the condition is met. Finally, if there is no sentence that fulfills this requirement, it will be included the one having the best cosine similarity with the query vector.
- Removal of a sentence: a sentence that exists in the summary is removed. The sentence $s_i \in S$ to be discarded must not damage the summary quality, so its cosine similarity with the query vector Q must be smaller than the average cosine similarity, i.e.:

$$cosim(s_i, Q) < \frac{1}{n}\sum_{j=1}^{n} cosim(s_j, Q). \tag{10}$$

  The sentence $s_i \in S$ is chosen randomly, being removed from the summary if it fulfills Eq. (10). Otherwise, another sentence $s_i \in S$ will be checked till it is fulfilled. Similarly, if there is no sentence that meets the requirement, the one with the smallest cosine similarity with the query vector will be removed.

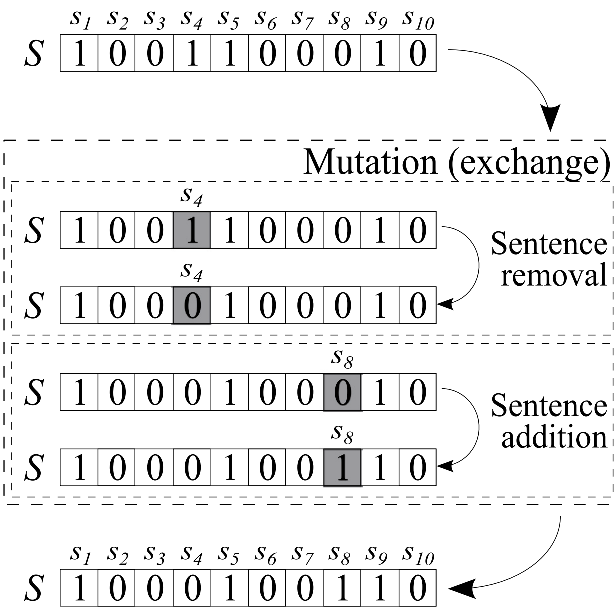


Fig. 1. Example of application of the mutation operator (exchange).

- Exchange of a sentence: a sentence from the summary is exchanged with another in the collection. The operation carried out in this case involves the removal of a sentence from the summary, followed by the addition of a new one.

An example of the mutation operator is presented in Fig. 1. The example shows a summary S that includes four sentences (out of the ten existing in the example collection, $n = 10$). In this case, the type of mutation corresponds to the exchange of a sentence, consisting firstly in the removal of a sentence, s_4, which is selected following Eq. (10) or for being the one with the smallest cosine similarity with the query vector. To complete the exchange, the addition of a different sentence is carried out. This sentence, s_8, is chosen considering Eq. (9) or being the one with the highest cosine similarity with the query vector. Finally, it can be observed how the application of the mutation operation has modified the summary S.

Secondly, the repair operation verifies that all summaries stored in the set of best solutions fulfill the condition related to the length constraint (Eq. (7)), and the summaries that do not meet it will be repaired. If the summary length is larger than the constraint, then it is repaired, whereas if it is smaller, then it is discarded since this situation occurs very infrequently.

The repair operator consists of the following. It deletes those sentences from the summary that contribute the least to its quality in terms of cosine similarity with the query vector Q. That is, for every sentence, the following repair score is computed as:

$$score^{rep}_{s_i} = \frac{cosim(s_i, Q)}{\frac{1}{n}\sum_{j=1}^{n} cosim(s_j, Q)}. \tag{11}$$

This score measures how similar a sentence is to the query, and it is used to assess the quality of each sentence in the summary. Therefore, the sentence with the lowest score, i.e., with the lowest similarity to the query, is removed from the summary. This process is performed again till the length constraint is met.

Fig. 2 shows a graphical example of the application of the repair operator. Here, the summary S contains six sentences from the example collection (which includes ten sentences, $n = 10$). As this summary exceeds the length restriction indicated in Eq. (7), it is necessary to repair it. To this end, as many sentences as necessary are removed

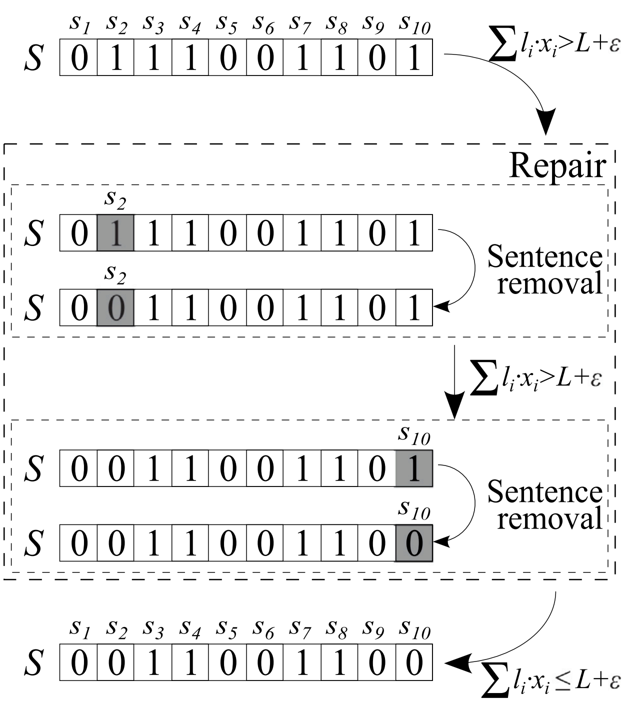


Fig. 2. Example of application of the repair operator.

until the length constraint is met. First, a sentence (s_2) is selected after computing the repair scores of all the sentences following Eq. (11), removing it for having the least cosine similarity with the query vector. Then, it is checked if the length constraint is satisfied. As it still does not meet it, this step is repeated and another sentence is chosen to be eliminated, s_{10} in this case. After removing this sentence, it is checked again if the summary fulfills the constraint, ending the repair operation when it finally has a valid length.

4.5. Reduction of the pareto set to a single solution

One of the characteristics of multi-objective optimization algorithms is that the obtained result consists of a set of non-dominated solutions, also called Pareto set. Any of these solutions is appropriate as the final solution, but, generally, only one is needed. For this reason, it is necessary to reduce the Pareto set to a single solution.

There are different types of methods to select a single solution from the Pareto set in the scientific literature. Specifically, different methods were analyzed and compared in [33], such as the consensus, hypervolume, the shortest distance to the ideal point (considering four distances), and the shortest distance to all points (considering five distances). In this work, all these automatic methods have been developed and evaluated in order to use the best one for this case. Further details about the eleven methods can be obtained from [33].

5. Experimental results

This section includes the evaluation metrics, datasets, parameter settings, results, ablation study, and comparisons with other approaches.

5.1. Evaluation metrics

The *Recall-Oriented Understudy for Gisting Evaluation* (ROUGE) metrics have been widely considered to evaluate the quality of the generated summaries [11]. ROUGE scores are the most used ones for this kind of text summarization. They assess the quality of a computer-generated summary by means of calculating the amount of overlapping units between it and a human-generated summary.

Table 2
Features of the TAC2009 datasets.

Description	Value
Number of topics	44
Number of documents per topic	10
Average number of sentences in a topic	154
Average number of terms in a topic	5513
Average number of different terms in a topic	939
Summary length constraint (words)	100

Table 3
Values tested in the experimental study for the specific parameters.

Parameter	Tested values
$sols_{max}$	100, 200, 250, 300, 400
ρ	1.1, 2, 3, 4, 5
$neigh_{max}$	3, 4, 5, 6, 7, 8, 9

Specifically, the considered ROUGE scores have been ROUGE-1, ROUGE-2, and ROUGE-SU4, which have been widely used in the scientific literature for the assessment of query-focused summaries. ROUGE-N is the N-gram recall between the system-generated summary and a set of human-generated summaries. When $N = 1$, it counts the number of unigram overlaps, and when $N = 2$, it calculates the number of bigram overlaps. ROUGE-SU4 counts the number of overlaps of skip-bigrams with a gap length equal to 4.

5.2. Datasets

For the experimentation, the datasets supplied by *Text Analysis Conference* (TAC) [34] have been used. TAC is an open standard benchmark from the National Institute of Standards and Technology (NIST, USA), very used for the assessment of query-focused summarization methods. Specifically, TAC2009 [10] has been considered to make comparisons with the competing approaches, since these datasets have been used in all these other approaches.

TAC2009 datasets include 44 topics. Every topic includes 10 documents and 4 human-generated summaries, which have been produced by human experts from NIST. Each summary is restricted to 100 words, and they are used as references to assess the quality of the computer-generated summaries. Table 2 contains some features of these datasets.

5.3. Parameter settings

The general parameters for IMOVNS are: the mutation probability, $p_m = 1/n$, as explained in Section 4.4; the maximum number of evaluations, $evals_{max} = 64000$, to make fair comparisons; and the scaling factor for the indicator-based fitness values, $\kappa = 0.05$, the same as originally proposed in [26]. An experimental study has been carried out for the specific parameters: the maximum number of saved solutions, $sols_{max}$; the reference point to calculate the hypervolume indicator, ρ; and the maximum number of neighborhoods, $neigh_{max}$. The tested values are presented in Table 3. The best configuration was: $sols_{max} = 250$, $\rho = 4$, and $neigh_{max} = 8$.

Regarding the method used for reducing the Pareto set to a single solution, the analysis carried out has reported that the consensus method produced the best results in ROUGE scores, both in average term and in number of topics. The consensus solution is formed from the Pareto set, that is, from the set of all non-dominated solutions. In this case, the consensus solution is composed with the most frequently used sentences in the summaries of the Pareto set, until the summary length constraint indicated in Eq. (7) is reached [33]. Note that this method has been used in all the experimentation of the proposed IMOVNS and in that of all competing multi-objective optimization approaches.

In order to ensure the statistical reliability of the results, 31 independent runs have been performed per every experiment, that is, per every

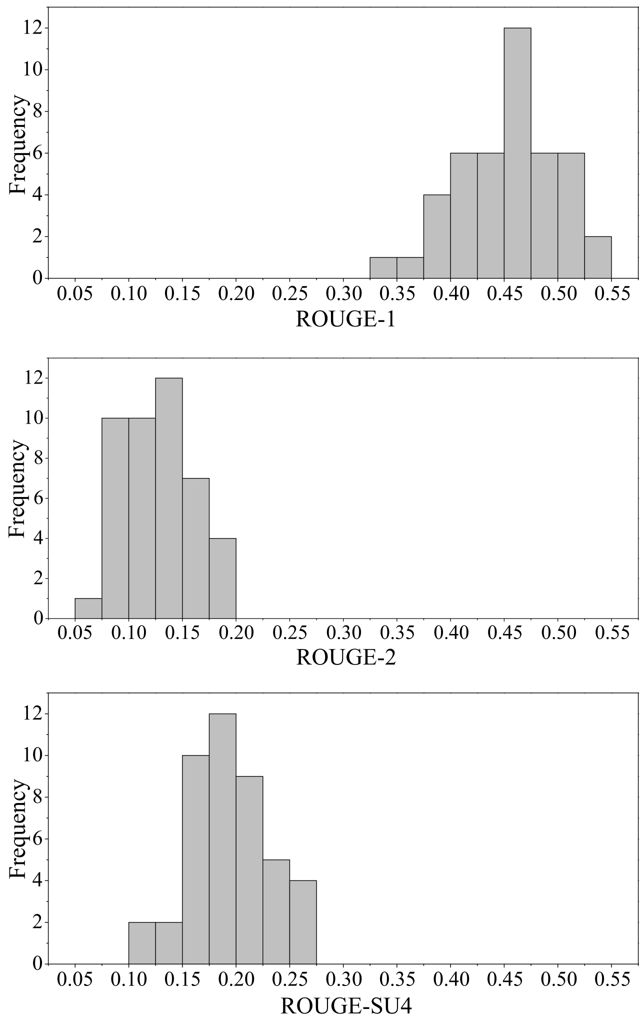


Fig. 3. Histograms provided by IMOVNS, based on the 31 independent runs performed for the 44 topics from TAC2009 datasets, for ROUGE-1, ROUGE-2, and ROUGE-SU4 scores.

topic from datasets. In addition, the experimentation has been carried out in a compute node with 4 processors AMD Opteron Abu Dhabi 6376 and with 96 GB RAM. The algorithm has been implemented and developed in C/C++ language in Eclipse Platform on Ubuntu 20.04.2.0 LTS (Focal Fossa).

5.4. Results of IMOVNS

Firstly, the results of ROUGE-1, ROUGE-2, and ROUGE-SU4 scores are statistically analyzed. Position and dispersion descriptive statistics are presented in Table 4, based on the 31 independent runs performed for the 44 topics from TAC2009 datasets.

As is reported in Table 4, mean scores are 0.452 for ROUGE-1, 0.129 for ROUGE-2, and 0.196 for ROUGE-SU4. They will be considered for comparison purposes in the next subsection.

The distribution of the ROUGE scores is represented in Fig. 3. This figure shows the histograms of ROUGE-1, ROUGE-2, and ROUGE-SU4 scores for the 44 topics. All of them approximate normal distributions. Note that ROUGE scores measure different types of N-grams (see Section 5.1), and therefore the histograms are not comparable to each other.

In addition, Fig. 4 shows the boxplots of ROUGE-1, ROUGE-2, and ROUGE-SU4 scores for the 44 topics. The median, the first quartile Q_1, and the third quartile Q_3 are represented as the central segment, the lower segment, and the upper segment of the box, respectively. Moreover, the minimum and maximum values are depicted as the lower whisker limit and the upper whisker limit, respectively. There are no outliers or extreme values.

Table 4
Descriptive statistics provided by IMOVNS, based on the 31 independent runs performed for the 44 topics from TAC2009 datasets, for ROUGE-1, ROUGE-2, and ROUGE-SU4 scores.

IMOVNS	ROUGE-1	ROUGE-2	ROUGE-SU4
Mean	0.452	0.129	0.196
Median	0.456	0.133	0.195
Standard deviation	0.047	0.032	0.036
Q_1	0.423	0.100	0.173
Q_3	0.486	0.150	0.222
Minimum	0.330	0.072	0.118
Maximum	0.537	0.196	0.271

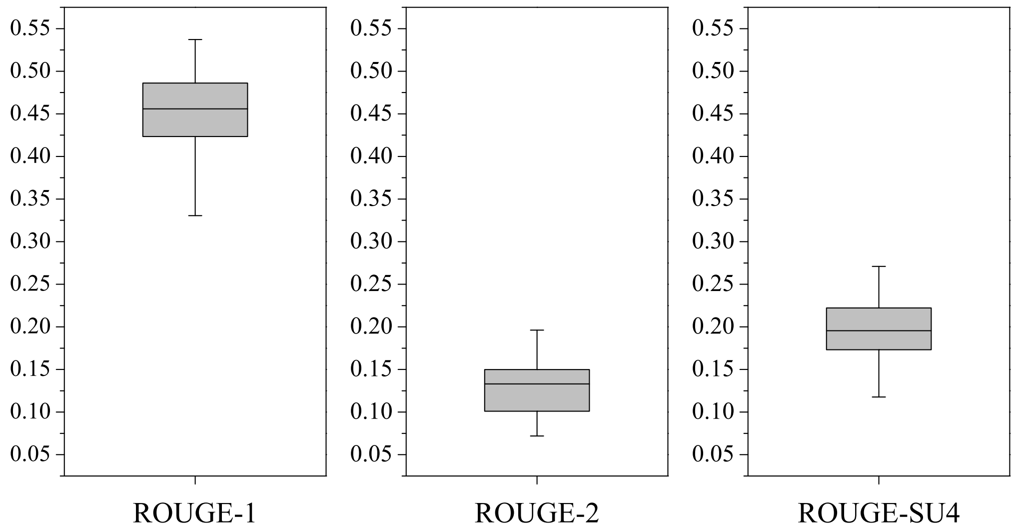
Table 5
Results provided by the ablation in the mutation operation (IMOVNS-AM) and of the tabu list (IMOVNS-AT) compared to the proposed IMOVNS for ROUGE-1, ROUGE-2, and ROUGE-SU4 scores (mean values).

Proposal	ROUGE-1	ROUGE-2	ROUGE-SU4
IMOVNS	**0.452**	**0.129**	**0.196**
IMOVNS-AM	0.422	0.118	0.180
IMOVNS-AT	0.415	0.114	0.176

5.5. Ablation study

In order to verify the contribution of the improvements introduced in the proposed IMOVNS with respect to the standard VNS, an ablation study has been carried out, consisting of the elimination of the problem-aware mutation operator and the list of tabu solutions, but not at the same time. The first component removed concerns the mutation operator, which has been designed specifically for the query-based multi-document summarization problem, replacing it with a standard mutation operator, such as the bitwise mutation used in NSGA-II [35]. This mutation flips a single random sentence, adding or removing it from the summary, so it does not consciously add quality to the mutated summary. As for the second component, it consists of eliminating the list of tabu solutions and not using it during the execution of the algorithm. Thus, the algorithm will not be able to know which solutions have already been used at the restart points, which will affect the search for solutions because it will not be possible to avoid using the same solutions again. We have named these versions as IMOVNS-AM (IMOVNS with Ablation in the Mutation) and IMOVNS-AT (IMOVNS with Ablation of the Tabu list). After detailing the components to be removed and evaluated in the ablation study, Table 5 shows the results, mean values after 31 independent runs, obtained for each of them and their comparison with the proposed IMOVNS.

The results reported in Table 5 show that, for the three ROUGE scores, the two removed components decrease the performance of IMOVNS, affecting the quality of the generated summaries. Besides, the contribution of both components can be measured in terms of percentage improvements in a separate way. Specifically, the problem-aware mutation operator provides percentage improvements of 7.19%, 10.04%, and 8.64% for ROUGE-1, ROUGE-2, and ROUGE-SU4 scores, respectively. As for the list of tabu solutions, its percentage improvements provided are 8.97% for ROUGE-1, 13.47% for ROUGE-2, and 11.39% for ROUGE-SU4 scores. Therefore, both components are essential in providing quality to the summaries generated by the proposed IMOVNS approach.



(a) Boxplot for ROUGE-1 scores. (b) Boxplot for ROUGE-2 scores. (c) Boxplot for ROUGE-SU4 scores.

Fig. 4. Boxplots provided by IMOVNS, based on the 31 independent runs performed for the 44 topics from TAC2009 datasets, for ROUGE-1, ROUGE-2, and ROUGE-SU4 scores.

Table 6
Comparison of IMOVNS with other proposals, based on the 31 independent runs performed for the 44 topics from TAC2009 datasets, for ROUGE-1, ROUGE-2, and ROUGE-SU4 scores (mean values ± standard deviation).

Proposal	ROUGE-1	ROUGE-2	ROUGE-SU4
IMOVNS	**0.452 ± 0.047**	**0.129 ± 0.032**	**0.196 ± 0.036**
MOSFLA [24]	0.440 ± 0.080	0.108 ± 0.042	0.173 ± 0.060
QOS-MOO-TFISF [23]	0.274 ± 0.041	0.095 ± 0.037	0.124 ± 0.030
QOS-MOO-WE [23]	0.267 ± 0.044	0.094 ± 0.037	0.110 ± 0.032
MEAD [14]	0.362 ± 0.084	0.094 ± 0.038	0.130 ± 0.032
LexRank [13]	0.365 ± 0.085	0.086 ± 0.035	0.127 ± 0.032

Table 7
Percentage improvements obtained by IMOVNS with respect to every proposal.

Proposal	ROUGE-1	ROUGE-2	ROUGE-SU4
MOSFLA [24]	2.73%	19.74%	13.06%
QOS-MOO-TFISF [23]	64.67%	35.95%	57.55%
QOS-MOO-WE [23]	69.24%	37.24%	77.37%
MEAD [14]	24.81%	37.04%	50.95%
LexRank [13]	23.96%	50.87%	53.68%
Average	**37.08%**	**36.17%**	**50.52%**

5.6. Comparison of IMOVNS with other proposals

This subsection presents the comparison of the results provided by IMOVNS with the ones from the proposals found in the scientific literature. All the proposals compared have performed the experimentation on the same datasets. In the first place, Table 6 shows the results of the competing algorithms with which the authors have been able to carry out the experimentation (those whose source code was obtained), in order to perform a more extensive comparative analysis as well as a statistical analysis. The mean values of ROUGE-1, ROUGE-2, and ROUGE-SU4 scores, in addition to the standard deviation, are presented and compared.

The results reported in Table 6 show that IMOVNS has achieved the best mean values for ROUGE-1, ROUGE-2, and ROUGE-SU4 scores. The percentage improvements, which range from 2.73% to 77.37%, are presented in Table 7. More specifically, the average improvement percentages obtained regarding these competing algorithms have been 37.08%, 36.17%, and 50.52% for ROUGE-1, ROUGE-2, and ROUGE-SU4 scores, respectively, showing important improvements in the three ROUGE scores.

Hypothesis tests have been applied to search for statistically significant differences between mean ROUGE scores. Applicability conditions (normality and homokedasticity) for ANOVA test could be assumed for the three ROUGE scores. In the three cases, ANOVA reported statistically significant differences, and Duncan tests were applied to search for differences between methods (IBM SPSS v27). Homogeneous subsets for Duncan test allow to divide methods into groups with closer mean values and to differentiate them from other groups. For ROUGE-1 score, there are three homogeneous subsets: the first one composed by QOS-MOO-WE and QOS-MOO-TFISF; the second one formed by MEAD and LexRank; and, finally, the third one formed by MOSFLA and IMOVNS. This last homogeneous subset reports an advantage in mean value for IMOVNS, and also a clear advantage in robustness with a coefficient of variation of 10.44% versus 18.20% for MOSFLA. In the case of ROUGE-2 score, IMOVNS formed alone a homogeneous subset, being better than all the other methods. IMOVNS also provides the best result with ROUGE-SU4 scores in a separate homogeneous subset.

Finally, it is remarkable that IMOVNS produces the most robust results in all ROUGE scores. Its coefficients of variation for ROUGE-1, ROUGE-2, and ROUGE-SU4 scores are 10.44%, 24.71%, and 18.57%, respectively, whereas the coefficients of variation of the other algorithms vary between 14.88% and 23.20% in ROUGE-1 score, between 38.66% and 40.67% in ROUGE-2 score, and between 24.17% and 34.61% in ROUGE-SU4 score. The results suggest that IMOVNS exhibits greater robustness compared to its competitors, as it yields better ROUGE scores that closely align with the increased means.

In the second place, Table 8 shows the results reported by the competing algorithms in their corresponding publications, since for these algorithms the source code was not available, and therefore the experimentation could not be carried out. In this case, only the mean values of ROUGE-1, ROUGE-2, and ROUGE-SU4 scores are compared, since these other proposals did not presented other statistical measures.

As can be appreciated in Table 8, the proposed IMOVNS has also achieved the best mean values for ROUGE-1, ROUGE-2, and ROUGE-SU4 scores. Finally, Table 9 shows the percentage improvements reached by IMOVNS regarding these other proposals. These

Table 8
Comparison of IMOVNS with other proposals for ROUGE-1, ROUGE-2, and ROUGE-SU4 scores (mean values). "NA" means that this score was not available.

Proposal	ROUGE-1	ROUGE-2	ROUGE-SU4
IMOVNS	**0.452**	**0.129**	**0.196**
BiProb EABS [22]	0.386	0.117	NA
W-LDA [17]	0.389	0.119	0.148
S-LDA [17]	0.390	0.121	0.150
S-sLDA [12]	0.390	0.122	0.149
Hiersum [19]	0.360	0.100	0.128
SVM [16]	0.365	0.103	0.132
Manifold [15]	0.371	0.101	0.134
KL-divergence [18]	0.347	0.082	0.112

Table 9
Percentage improvements obtained by IMOVNS with respect to every proposal. "NA" means that this percentage was not available.

Proposal	ROUGE-1	ROUGE-2	ROUGE-SU4
BiProb EABS [22]	17.25%	10.24%	NA
W-LDA [17]	16.17%	8.49%	31.98%
S-LDA [17]	15.84%	6.96%	30.57%
S-sLDA [12]	15.81%	5.74%	31.45%
Hiersum [19]	25.59%	28.80%	52.81%
SVM [16]	23.87%	25.79%	48.29%
Manifold [15]	21.74%	27.53%	45.75%
KL-divergence [18]	30.34%	57.70%	75.10%
Average	**20.83%**	**21.41%**	**45.13%**

percentages range from 5.74% to 75.10%, obtaining average percentages of 20.83% for ROUGE-1, 21.41% for ROUGE-2, and 45.13% for ROUGE-SU4 scores, also showing important improvements in the three ROUGE scores. Therefore, it can be concluded that IMOVNS has outperformed all the proposals found in the scientific literature.

6. Conclusions

The problem of query-focused summarization has become relevant in recent years. This kind of summarization is unique in that it requires user information to produce an automatic summary, being this information a query given. In addition to the query relevance, the redundancy reduction is also a relevant aspect, as these are the two most commonly used criteria in query-focused summarization.

An Indicator-based Multi-Objective Variable Neighborhood Search (IMOVNS) algorithm has been designed, implemented, and tested for the first time for the resolution of the query-focused extractive multi-document summarization problem. The adjustment of this metaheuristic method, based on VNS algorithm, as an indicator-based multi-objective optimization approach has led to a substantial improvement of the results. This has been possible due to the several advances developed, such as the restarting process, the list of tabu solutions, and the set with the best solutions found. In addition, the mutation operator and the repair operation have also been specifically designed for this task.

Regarding the results obtained, the analysis has reported that IMOVNS is a competitive approach outperforming the state-of-the-art. Large percentage improvements ranging from 2.73% to 69.24% in ROUGE-1 score, from 5.74% to 57.70% in ROUGE-2 score, and from 13.06% to 77.37% in ROUGE-SU4 score show that this approach should be considered for future applications.

As a future work, the proposed indicator-based multi-objective optimization approach will be adapted to solve the comment-based multi-document summarization problem. This kind of summarization takes into account the comments that users post on online documents, such as blogs, news websites, or social networks, as they can be helpful for recognizing their relevant points in a collaborative way. This involves different objective functions in a different but related problem that could be benefited from an adaptation of the proposed algorithm.

CRediT authorship contribution statement

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

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.

Data availability

The authors do not have permission to share data.

Acknowledgments

This research was supported by Ministry of Science, Innovation and Universities - Spain and State Research Agency - Spain (Projects PID2019-107299GB-I00, PID2021-122209OB-C32, and RED2022-134540-T funded by MICIU/AEI/10.13039/501100011033), Junta de Extremadura - Spain (Projects GR21017 and GR21057), and European Union (European Regional Development Fund). Jesus M. Sanchez-Gomez was supported by Junta de Extremadura and European Union (European Social Fund) under the doctoral fellowship PD18057.

References

[1] H. Hassani, C. Beneki, S. Unger, M.T. Mazinani, M.R. Yeganegi, Text mining in big data analytics, Big Data Cogn. Comput. 4 (1) (2020) 1, http://dx.doi.org/10.3390/bdcc4010001.
[2] W.S. El-Kassas, C.R. Salama, A.A. Rafea, H.K. Mohamed, Automatic text summarization: A comprehensive survey, Expert Syst. Appl. 165 (2021) 113679, http://dx.doi.org/10.1016/j.eswa.2020.113679.
[3] A. John, P.S. Premjith, M. Wilscy, Extractive multi-document summarization using population-based multicriteria optimization, Expert Syst. Appl. 86 (2017) 385–397, http://dx.doi.org/10.1016/j.eswa.2017.05.075.
[4] H. Van Lierde, T.W.S. Chow, Learning with fuzzy hypergraphs: A topical approach to query-oriented text summarization, Inform. Sci. 496 (2019) 212–224, http://dx.doi.org/10.1016/j.ins.2019.05.020.
[5] M. Gambhir, V. Gupta, Recent automatic text summarization techniques: A survey, Artif. Intell. Rev. 47 (1) (2017) 1–66, http://dx.doi.org/10.1007/s10462-016-9475-9.
[6] J.-g. Yao, X. Wan, J. Xiao, Recent advances in document summarization, Knowl. Inf. Syst. 53 (2) (2017) 297–336, http://dx.doi.org/10.1007/s10115-017-1042-4.
[7] R.M. Alguliev, R.M. Aliguliyev, C.A. Mehdiyev, Sentence selection for generic document summarization using an adaptive differential evolution algorithm, Swarm Evol. Comput. 1 (4) (2011) 213–222, http://dx.doi.org/10.1016/j.swevo.2011.06.006.
[8] J.M. Sanchez-Gomez, M.A. Vega-Rodríguez, C.J. Pérez, Extractive multi-document text summarization using a multi-objective artificial bee colony optimization approach, Knowl.-Based Syst. 159 (2018) 1–8, http://dx.doi.org/10.1016/j.knosys.2017.11.029.
[9] P. Hansen, N. Mladenović, J.A.M. Pérez, Variable neighbourhood search: Methods and applications, Ann. Oper. Res. 175 (1) (2010) 367–407, http://dx.doi.org/10.1007/s10479-009-0657-6.
[10] H.T. Dang, K. Owczarzak, Overview of the TAC 2009 summarization track, in: Text Analysis Conference, TAC, National Institute of Standards and Technology (NIST), 2009, pp. 111–113.
[11] C.-Y. Lin, ROUGE: A package for automatic evaluation of summaries, in: Text Summarization Branches Out: Proceedings of the ACL-04 Workshop, Vol. 8, Association for Computational Linguistics, 2004, pp. 74–81.
[12] J. Li, S. Li, A novel feature-based bayesian model for query focused multi-document summarization, Trans. Assoc. Comput. Linguist. 1 (2013) 89–98, http://dx.doi.org/10.1162/tacl_a_00212.
[13] G. Erkan, D.R. Radev, LexRank: Graph-based lexical centrality as salience in text summarization, J. Artificial Intelligence Res. 22 (2004) 457–479, http://dx.doi.org/10.1613/jair.1523.

[14] D.R. Radev, H. Jing, M. Styś, D. Tam, Centroid-based summarization of multiple documents, Inf. Process. Manage. 40 (6) (2004) 919–938, http://dx.doi.org/10.1016/j.ipm.2003.10.006.
[15] X. Wan, J. Yang, J. Xiao, Manifold-ranking based topic-focused multi-document summarization, in: Proceedings of the 20th International Joint Conference on Artifical Intelligence, IJCAI, Vol. 7, 2007, pp. 2903–2908.
[16] L. Li, K. Zhou, G.-R. Xue, H. Zha, Y. Yu, Enhancing diversity, coverage and balance for summarization through structure learning, in: Proceedings of the 18th International Conference on World Wide Web, ACM, 2009, pp. 71–80, http://dx.doi.org/10.1145/1526709.1526720.
[17] Y. Li, S. Li, 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, 2014, pp. 1197–1207.
[18] C.-Y. Lin, G. Cao, J. Gao, J.-Y. Nie, 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, Association for Computational Linguistics, 2006, pp. 463–470, http://dx.doi.org/10.3115/1220835.1220894.
[19] A. Haghighi, L. Vanderwende, 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, Association for Computational Linguistics, 2009, pp. 362–370.
[20] L. Marujo, R. Ribeiro, D.M. de Matos, J.P. Neto, A. Gershman, J. Carbonell, Extending a single-document summarizer to multi-document: A hierarchical approach, in: Proceedings of the Fourth Joint Conference on Lexical and Computational Semantics, Association for Computational Linguistics, 2015, pp. 176–181, http://dx.doi.org/10.18653/v1/S15-1020.
[21] L. Marujo, W. Ling, R. Ribeiro, A. Gershman, J. Carbonell, D.M. de Matos, J.P. Neto, Exploring events and distributed representations of text in multi-document summarization, Knowl.-Based Syst. 94 (2016) 33–42, http://dx.doi.org/10.1016/j.knosys.2015.11.005.
[22] A. Bossard, C. Rodrigues, An evolutionary algorithm for automatic summarization, in: Proceedings of the International Conference on Recent Advances in Natural Language Processing, RANLP, 2017, pp. 111–120, http://dx.doi.org/10.26615/978-954-452-049-6_017.
[23] Y. Fors-Isalguez, J. Hermosillo-Valadez, M. Montes-y-Gómez, Query-oriented text summarization based on multiobjective evolutionary algorithms and word embeddings, J. Intell. Fuzzy Systems 34 (5) (2018) 3235–3244, http://dx.doi.org/10.3233/JIFS-169506.
[24] J.M. Sanchez-Gomez, M.A. Vega-Rodríguez, C.J. Pérez, A multi-objective memetic algorithm for query-oriented text summarization: Medicine texts as a case study, Expert Syst. Appl. 198 (2022) 116769, http://dx.doi.org/10.1016/j.eswa.2022.116769.
[25] G. Salton, C. Buckley, Term-weighting approaches in automatic text retrieval, Inf. Process. Manage. 24 (5) (1988) 513–523, http://dx.doi.org/10.1016/0306-4573(88)90021-0.
[26] E. Zitzler, S. Künzli, Indicator-based selection in multiobjective search, in: International Conference on Parallel Problem Solving from Nature, Springer, 2004, pp. 832–842, http://dx.doi.org/10.1007/978-3-540-30217-9_84.
[27] E. Rahimian, K. Akartunalı, J. Levine, A hybrid integer programming and variable neighbourhood search algorithm to solve nurse rostering problems, European J. Oper. Res. 258 (2) (2017) 411–423, http://dx.doi.org/10.1016/j.ejor.2016.09.030.
[28] E. de Vargas Fortes, L.H. Macedo, P.B. de Araujo, R. Romero, A VNS algorithm for the design of supplementary damping controllers for small-signal stability analysis, Int. J. Electr. Power Energy Syst. 94 (2018) 41–56, http://dx.doi.org/10.1016/j.ijepes.2017.06.017.
[29] D. Schermer, M. Moeini, O. Wendt, A hybrid VNS/Tabu search algorithm for solving the vehicle routing problem with drones and en route operations, Comput. Oper. Res. 109 (2019) 134–158, http://dx.doi.org/10.1016/j.cor.2019.04.021.
[30] J. Pei, X. Liu, W. Fan, P.M. Pardalos, S. Lu, A hybrid BA-VNS algorithm for coordinated serial-batching scheduling with deteriorating jobs, financial budget, and resource constraint in multiple manufacturers, Omega 82 (2019) 55–69, http://dx.doi.org/10.1016/j.omega.2017.12.003.
[31] J. Li, ROUGE metric, 2020, https://pypi.org/project/rouge-metric/, (Last Accessed 15 July 2024).
[32] M. Porter, The porter stemming algorithm, 2023, http://www.tartarus.org/martin/PorterStemmer/, (Last Accessed 15 July 2024).
[33] J.M. Sanchez-Gomez, M.A. Vega-Rodríguez, C.J. Pérez, Comparison of automatic methods for reducing the Pareto front to a single solution applied to multi-document text summarization, Knowl.-Based Syst. 174 (2019) 123–136, http://dx.doi.org/10.1016/j.knosys.2019.03.002.
[34] Text analysis conference, 2024, https://tac.nist.gov/, (Last Accessed 15 July 2024).
[35] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II, IEEE Trans. Evol. Comput. 6 (2) (2002) 182–197, http://dx.doi.org/10.1109/4235.996017.