

Automatic update summarization by a multi-objective number-one-selection genetic approach

Jesus M. Sanchez-Gomez, Miguel A. Vega-Rodríguez, and Carlos J. Pérez

Abstract—Currently, the explosive growth of the information available on the internet makes automatic text summarization systems increasingly important. A particularly relevant challenge is the update summarization task. Update summarization differs from traditional summarization in its dynamic nature. While traditional summarization is static, i.e., the document collections about a specific topic remain unchanged, update summarization addresses dynamic document collections based on a specific topic. Therefore, update summarization consists of summarizing the new document collection under the assumption that the user has already read a previous summarization and only the new information is interesting. The Multi-Objective Number-One-selection Genetic Algorithm (MONOGA) has been designed and implemented to address this problem. The proposed algorithm produces a summary that is relevant to the user's given query, and it also contains updates information. Experiments were conducted on Text Analysis Conference (TAC) datasets, and Recall-Oriented Understudy for Gisting Evaluation (ROUGE) metrics were considered to assess the model performance. The results obtained by the proposed approach outperform those from the existing approaches in the scientific literature, obtaining average percentage improvements between 12.74% and 55.03% in the ROUGE scores.

Index Terms—Update summarization, Multi-objective optimization, Genetic algorithm, Number-One selection, MONOGA, TAC.

I. INTRODUCTION

AUTOMATIC text summarization has become a very relevant research field due to the growth of the information on the Worldwide Web. In addition, internet users demand access to the most valuable information about a certain topic. Using text mining tools, it is possible to retrieve important information from a large document collection [1]. In particular, these tools are able to generate a summary with the desired information [2].

One of the most challenging summarization problems is update summarization; it is the natural evolution of traditional summarization. An update summarization produces a summary from a newer document collection with respect to another older document collection, i.e., update summarization deals

with dynamic document collections [3]. The objective is to produce a summary from the current document collection under the assumption that the user has already read the old document collection. Therefore, the user is only interested in the new information. Moreover, this summary should be relevant to the query provided by the user [4], as opposed to containing the generic content of a conventional summarization [5]. The intended summary is extractive, which means that the generated summary is formed by sentences that exist in the document collection [6] and is therefore different from an abstractive summarization [7]. In addition, since an update summarization works with multiple documents, it is also defined as a multidocument summarization method [8]. In the scientific literature, most automatic text summarization methods have been formulated as single-objective approaches, in which all the criteria to be optimized are included in a single objective function (see, e.g., [9]). However, the criteria must be weighted by a subjective assignment, which affects the final solution. In recent times, multi-objective approaches have gained relevance because they do not need subjective weights, optimize several objective functions simultaneously and have a good performance [10]. To the best of the authors' knowledge, the update summarization problem has never been approached from a multi-objective optimization point of view.

In this paper, the update summarization problem has been addressed with a multi-objective optimization approach. The multi-objective number-one-selection genetic algorithm (MONOGA) has been designed, implemented and tested to solve this problem. Query relevance, redundancy reduction and updating have been simultaneously optimized as objective functions. The experiments have been carried out with datasets from the Text Analysis Conference (TAC) [11], [12]. Recall-Oriented Understudy for Gisting Evaluation (ROUGE) metrics have been used for model performance [13]. The main research contributions of this paper are as follows:

- For the first time, the update summarization problem is formulated from a multi-objective optimization point of view.
- Three criteria are considered for optimization, they are, query relevance, redundancy reduction and updating.
- The Multi-Objective Number-One-selection Genetic Algorithm (MONOGA) is developed to solve the update summarization problem.
- An original selection operator based on number-one selection is specifically designed for the proposed algo-

Jesus M. Sanchez-Gomez and Miguel A. Vega-Rodríguez are with Universidad de Extremadura (<https://ror.org/0174shg90>), Departamento de Tecnología de Computadores y Comunicaciones, Campus Universitario s/n, 10003 Cáceres, Spain.
E-mail: jmsanchezgomez@unex.es, mavega@unex.es.

Carlos J. Pérez is with Universidad de Extremadura (<https://ror.org/0174shg90>), Departamento de Matemáticas, Campus Universitario s/n, 10003 Cáceres, Spain.
E-mail: carper@unex.es.

Corresponding author: Jesus M. Sanchez-Gomez.

rithm.

- The remaining operators, crossover, mutation and repair are specifically problem-aware developed.
- The obtained experimental results show that this approach outperforms the existing approaches in the scientific literature.

The rest of the paper is organized as follows. Section II presents the related work. Section III describes the mathematical definition of the optimization problem. Section IV explains in detail the proposed method, including the proposed operators. Section V contains the datasets, the evaluation metrics, the experimental settings, the results obtained, the contribution of its operators and a comparison with other approaches. Finally, conclusions and future research are presented in Section VI.

II. RELATED WORK

In this section, the update summarization approaches are reviewed in chronological order. First, an extractive approach based on manifold ranking with sink points (MRSP) for update summarization was proposed in [14]. This approach also leverages a manifold ranking process over the sentence manifold to find the relevant sentences. Moreover, based on the intrinsic sentence manifold with sink points, the ranking process can further capture the novelty and diversity of the update summary. In [15], a document summarization approach based on information distance was proposed. With the aim of addressing update summarization, information distance theory is extended to the conditional information distance among many objects. In this way, the best summary, formed with sentences from the new document collection, is the one with the minimum information distance to the document collection. A principled and versatile framework for multidocument summarization using the minimum weighted dominating set (MWDS) was proposed in [16]. This framework is modeled for summarization tasks, such as update summarizations, by using a sentence graph generated from the input documents. The generated summary is based on the graph domination idea. In [17], a method based on an incremental hierarchical clustering framework for update summarization (IHCUS) was proposed. This technique produces update summaries by reorganizing the sentence clusters immediately after new documents or sentences arrive so that the corresponding summaries are efficiently updated. In addition, the hierarchical relationships among the sentences are also displayed and reconstructed in real time.

Later, a graph-based regularization marginal framework (MarginRank) for update summarization was proposed in [18]. It finds sentences that have the maximal marginal relevance to the topic, considering that the sentences covered by the past documents are nonrelevant. The cost function consists of manifold regularization terms and topic-relevance fitting terms. In [19], a combination of a multidocument update summarization system, the Clustering Based Sentence Extractor for Automatic Summarization (CBSEAS), with a genetic algorithm was presented. This system contains a method for detecting redundancy to produce summaries with good informational

diversity. In addition, the use of a genetic algorithm aids in finding the best suited parameter combination.

A graph-ranking-based method for update summarization, the quadratic constrained quadratic programming summarization (QCQPSum) optimization method, was proposed in [20]. This method carries out constrained reinforcements on a sentence graph to determine the salience of the sentences. The objective function of QCQPSum contains the criteria of salience, relevance, nonredundancy and update. In [21], a selection framework based on evolutionary manifold ranking with a redundancy removal strategy with exponential decay and normalized spectral clustering (EEDS) was proposed. This framework captures the temporal characteristics and relay propagation of information in the dynamic data stream (characteristics of update summarization) and the user's needs. Moreover, it tries to keep the summary content impactful, novel and relevant to the topic. In addition, it considers normalized spectral clustering to ensure that the content of the summary has a high coverage for each subtopic.

A modification of the integer linear programming (ILP) problem that used discriminative models to measure the salience and novelty of sentences from documents was proposed in [22]. Supervised models and a rich set of features are used to learn the weights for the bigram concepts used in the ILP model; then, a sentence reranking component is designed to score the summary candidate sentences generated by the ILP model. Afterward, in [23], an update summarization framework was developed by including nonredundancy features in a more explicit way by means of semantic sentence similarities. The summarization framework is based on ILP, and it consists of two main steps. First, semantic clustering over the input sentences of the documents is performed, including the old and new document sets. The network-based clustering algorithm chosen for simulating the flow in graphs is the Markov CLuster algorithm (MCL). Second, a subset of sentences is selected for the summary while considering the semantic information resulting from the clustering step.

An Internet-based Multi-Document Dynamic Summarization (IMDDS) framework was introduced in [24]. It is composed of three main modules: The feature extraction module, which uses a keyword extraction algorithm for measuring the sentence historical and sentence salience; the information filtering module for sorting the sentences according to these features and filtering the historical information; and the sentence weighting module, which uses a manifold ordering algorithm for scoring.

In [25], a versatile framework based on game theory was proposed. This framework is integrated with the Wikipedia ontology, and it exploits the submodularity among the sentences of the underlying documents. This submodularity is optimized using a Shapley value-based algorithm to enhance the summarization performance by identifying the relevant sentences.

Finally, several techniques for inferring background knowledge from summarization data were developed in [26]. The KLearn framework includes scoring functions that explicitly model background knowledge. In addition, this framework is based on the information selection model, which proposes that

the information in the summary should have high relevance (the summary covers the main information), low redundancy (the information in the summary is diverse) and high informativeness (the summary contains new information).

All the reviewed approaches experimented with the TAC datasets. In particular, the TAC2008 datasets were used in all of them, and the TAC2009 datasets were used in 7 out of the 13. For the evaluation metrics, all the approaches used the ROUGE scores. Specifically, ROUGE-2 and ROUGE-SU4 were the most commonly used scores. The ROUGE-1 and ROUGE-L scores were also used.

III. PROBLEM DEFINITION

In the scientific literature, the summarization problem has been formulated using vector-based word approaches, where sentences are represented as vectors of words. The update summarization problem involves the use of two document collections, one older or previous, and another newer or current. In addition, a query-focused summary from the previous document collection must also be provided as a reference to assess the novelty of the update summary to be generated.

First, let $D = \{d_1, d_2, \dots, d_N\}$ be the new or current document collection with N documents. The document collection D can also be represented as a set of n sentences as $D = \{s_1, s_2, \dots, s_n\}$. In addition, let $T = \{t_1, t_2, \dots, t_m\}$ be a set with all the m different terms in D . In this way, a single sentence can be symbolized as a m -dimensional vector as $s_i = (w_{i1}, w_{i2}, \dots, w_{im})$ in which every element is related to the weight of the term t_k in the sentence s_i . One way to calculate this weight w_{ik} is by using the term-frequency inverse-sentence-frequency (TF-ISF) scheme [27], which is formulated as:

$$w_{ik} = tf_{ik} \cdot \log\left(\frac{n}{n_k}\right), \quad (1)$$

where tf_{ik} is the term frequency, and refers to the number of occurrences of the term t_k in the sentence s_i , and $\log(n/n_k)$ is the inverse sentence frequency, with n being the number of sentences, and n_k is the number of sentences in which the term t_k occurs.

The most commonly used similarity criterion between pairs of sentences is the cosine similarity measure. It computes the resemblance between a pair of sentences $s_i = (w_{i1}, w_{i2}, \dots, w_{im})$ and $s_j = (w_{j1}, w_{j2}, \dots, w_{jm})$ as:

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

The query given by the user can also be defined in the same way as a sentence: $Q = (q_1, q_2, \dots, q_m)$, where each element q_k corresponds to the weight of the corresponding k -th term.

Once the notation and initial formulation have been described, the formulation of the optimization problem is presented. The goal is to produce a summary $S \subset D$ taking into account the following points:

- Query relevance. The summary must include the most relevant sentences according to the query given by the user.
- Redundancy reduction. The summary must not contain redundant sentences.
- Update. The summary only should gather the sentences that contain new information that the user has not previously read.
- Length constraint. The summary must have a predefined fixed length.

A binary decision variable is defined: $x_i \in \{0, 1\}$. It takes into account the presence or absence ($x_i = 1$ or $x_i = 0$) of the i -th sentence in the summary, so the solution representation or the decision vector is denoted as $X = (x_1, x_2, \dots, x_n)$.

The first objective function is related to the query relevance criterion, $\Phi_{QR}(X)$. The query relevance is defined as the cosine similarity between every sentence in the summary, $s_i \in S$, and the query vector Q . Thus, the following objective function, which was first introduced in [28], must be maximized:

$$\Phi_{QR}(X) = \sum_{i=1}^n \text{cosim}(s_i, Q) \cdot x_i. \quad (3)$$

The second objective function refers to the redundancy reduction criterion, $\Phi_{RR}(X)$. The redundancy is measured as the sum of the cosine similarities between every pair of sentences in the summary, $s_i, s_j \in S$. In this case, the simultaneous presence of the pair of sentences s_i and s_j is determined with $x_i x_j$. As redundancy reduction is the opposite concept, the objective function expressed below (first introduced in [29]) should be maximized:

$$\Phi_{RR}(X) = - \sum_{i=1}^{n-1} \sum_{j=i+1}^n \text{cosim}(s_i, s_j) \cdot x_i x_j. \quad (4)$$

The third objective function concerns the update criterion, $\Phi_{UP}(X)$. This criterion is established as the opposite of the total of the maxima of the cosine similarities between each sentence in the summary, $s_i \in S$, and each sentence of the reference summary, $s_j^P \in S^P$. This reference summary S^P is the summary of the previous document collection D^P , where D^P has a total of n^P sentences, and the ones included in S^P have $x_j^P = 1$. From now on, the super index P refers to the previous or old document collection. In conclusion, the following objective function must be maximized:

$$\Phi_{UP}(X) = - \sum_{i=1}^n \max_{1 \leq j \leq n^P} \text{cosim}(s_i, s_j^P) \cdot x_j^P \cdot x_i. \quad (5)$$

Finally, once the three objective functions have been defined, the multi-objective update summarization problem is formulated as follows:

$$\max \phi(X) = \{\Phi_{QR}(X), \Phi_{RR}(X), \Phi_{UP}(X)\}, \quad (6)$$

$$\text{subject to } L - \varepsilon \leq \sum_{i=1}^n l_i \cdot x_i \leq L + \varepsilon, \quad (7)$$

where L is the predefined fixed length for the summary, l_i is the length of the i -th sentence and ε is the length tolerance computed as:

$$\varepsilon = \max_{1 \leq i \leq n} l_i - \min_{1 \leq i \leq n} l_i. \quad (8)$$

Equations 7 and 8 were first introduced in [30].

IV. MULTI-OBJECTIVE NUMBER-ONE-SELECTION GENETIC ALGORITHM (MONOGA)

This section is devoted to presenting the proposed method. First, its operators of selection, crossover, mutation and repair are detailed. All of them have been specifically designed and developed for this proposal. Then, the main steps of the algorithm are described. Finally, the reduction of the Pareto front to one solution (summary) is explained.

A. Number-One selection operator

In genetic algorithms, there are various methods to select the individuals who will be crossed. In particular, one of the most commonly used methods is a binary tournament selection, which is used by NSGA-II. This method involves running a binary tournament for choosing every parent, $Parent1$ and $Parent2$. For each parent, two individuals from the population are randomly selected for the binary tournament, and the one with the higher quality will be chosen for the crossover. This technique ensures both the diversity and the quality of the future offspring solution.

For the proposed algorithm (MONOGA), a novel selection operator has been developed, the number-one-based selection. This selection considers that the first parent of the offspring solution will always be the best individual of the population. At the time of the selection, it is assumed that the population is sorted by Pareto rank and crowding; therefore, the individual placed in the first position (with the highest quality value) will be selected as the first parent. The second parent will be the individual of the population selected by an index (an input parameter of this selection operator). This index can be generated in an iterative way. Therefore, every individual from the population will be crossed with the best one, the number one. This operator guarantees very high-quality offspring solutions. Algorithm 1 explains this selection procedure.

Algorithm 1 Number-One based selection operator pseudocode.

Input: Pop : Population of solutions, $index$: index for the second parent.

Output: $Parent1$ and $Parent2$: Individuals selected as the first and second parent.

- 1: $Parent1 \leftarrow Pop[1]$
 - 2: $Parent2 \leftarrow Pop[index]$
 - 3: **return** $Parent1, Parent2$
-

B. Problem-aware crossover operator

Given the two previously selected parents, the crossover operator produces an offspring solution in a determined way. Different crossover techniques exist. One of the most widely used is the single-point crossover, which is used by NSGA-II. This kind of crossover randomly chooses a single point (the crossover point) on both parent solutions, dividing them into two parts. The offspring solution will have the first part of its solution from the first parent and the second part from the

second parent. In this way, the offspring solution will have genetic information from its two parents.

The crossover operator developed for MONOGA has been specifically designed as a problem-aware operator. This method generates an offspring solution from the best sentences of its two parents. First, a set of candidate sentences, CS , is generated by joining the sentences used in both parents, i.e., $CS \leftarrow Parent1 \cup Parent2$. Then, the sentences of the candidate set are sorted in descending order according to a crossover score, $score_{s_i}^{cro}$. This score takes into account, for each sentence $s_i \in CS$, both the cosine similarity between the sentence and the query Q and the update between the sentence and the previous or old reference summary S^P as follows:

$$score_{s_i}^{cro} = \frac{cosim(s_i, Q)}{\frac{1}{n} \sum_{j=1}^n cosim(s_j, Q)} + \frac{update(s_i, S^P)}{\frac{1}{n} \sum_{j=1}^n update(s_j, S^P)}, \quad (9)$$

where $update(s_i, S^P)$ is calculated as:

$$update(s_i, S^P) = - \max_{1 \leq j \leq n^P} cosim(s_i, s_j^P) \cdot x_j^P. \quad (10)$$

Finally, the offspring solution (summary) is formed by adding the best sentences (those with the highest $score_{s_i}^{cro}$ scores) until reaching the maximum length constraint determined in Equation 7.

C. Problem-aware mutation operator

A mutation operator focuses on modifying an individual to keep the genetic diversity in the population. There are several techniques to address a mutation. The bitwise mutation is one of the most widespread methods and is used in NSGA-II. It flips a single random bit of the individual according to a defined mutation probability.

As in the previous operators, the mutation operator developed for the proposed MONOGA has been designed to be problem aware. The proposed mutation operator relies in adding, removing, or exchanging a single sentence from the summary S to improve its quality. Any of these three alternatives has the same probability of being performed in the mutation operation, considering that the length constraint (Equation 7) should be fulfilled. In addition, the mutation probability, mut_{prob} , is set to $1/n$, so a single sentence is always mutated. Therefore, this operation can be performed in any of the following ways:

- 1) Adding a sentence. A random sentence that is not in the summary, $s_i \notin S$, will be added. It must improve its quality in terms of cosine similarity with the query Q , and in terms of updating with the reference summary S^P . These two conditions can be formulated as follows:

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

$$update(s_i, S^P) > \frac{1}{n} \sum_{j=1}^n update(s_j, S^P). \quad (11)$$

If these conditions are fulfilled, then the sentence is included in the summary. Otherwise, these conditions

are examined for the remaining sentences that are not in the summary until one of them meets these conditions. This sentence will be included in the summary. If no sentence satisfies the conditions, the following mutation score, $score_{s_i}^{mut}$, is computed for each sentence:

$$score_{s_i}^{mut} = \frac{\cosim(s_i, Q) - \frac{1}{n} \sum_{j=1}^n \cosim(s_j, Q)}{\frac{1}{n} \sum_{j=1}^n \cosim(s_j, Q)} + \frac{update(s_i, S^P) - \frac{1}{n} \sum_{j=1}^n update(s_j, S^P)}{\frac{1}{n} \sum_{j=1}^n update(s_j, S^P)}. \quad (12)$$

Finally, the sentence with the largest mutation score will be added to the summary.

- 2) Removing a sentence. In this case, a random sentence in the summary, $s_i \in S$, will be removed. The removed sentence should not deteriorate the quality of the summary, both in terms of cosine similarity with the query, and in terms of updating with the reference summary. These requirements can be defined as follows:

$$\begin{aligned} \cosim(s_i, Q) &< \frac{1}{n} \sum_{j=1}^n \cosim(s_j, Q), \\ update(s_i, S^P) &< \frac{1}{n} \sum_{j=1}^n update(s_j, S^P). \end{aligned} \quad (13)$$

In the same way as before, if the requirements are met, then the sentence is removed from the summary. Otherwise, the requirements are checked for the remaining sentences from the summary as long as a sentence fulfills these conditions. This sentence will be removed from the summary. If no sentence in the summary fulfills these requirements, then the mutation score formulated in Equation 12 is calculated for all the sentences, and the sentence with the smallest score is removed.

- 3) Exchanging a sentence. The third alternative replaces a sentence in the summary with another not included in it. For this task, the operation performed is, first, removing a sentence, and then, adding a distinct one, following the previous steps.

D. Problem-aware repair operator

As previously explained, crossover and mutation operators take into account the length constraint. However, they use an estimation (maximum and minimum number of sentences in a summary). As a result, only a very reduced number of solutions violate this constraint. Regardless, an operator is necessary to fix the solutions that do not satisfy the length constraint in Equation 7. For this reason, a problem-aware repair operator has been designed for this task. It must repair the summary, deteriorating its quality as little as possible, in terms of cosine similarity with the query Q , and in terms of updating with the reference summary S^P .

The repair operation is carried out as follows. The exact length of the summary is checked. If it is greater than the maximum length constraint, $L + \varepsilon$, then the solution is repaired. On the other hand, if the length of the summary is less than the minimum length constraint, $L - \varepsilon$, then the solution is rejected because this happens with a very low frequency.

Therefore, when a solution is repaired, one or more sentences from the summary that infringes on the maximum length constraint, S^* , must be removed. For this, a repair score, $score_{s_i}^{rep}$, is calculated for every sentence s_i in the summary as follows:

$$score_{s_i}^{rep} = \frac{\cosim(s_i, Q)}{\frac{1}{n} \sum_{j=1}^n \cosim(s_j, Q)} + \frac{update(s_i, S^P)}{\frac{1}{n} \sum_{j=1}^n update(s_j, S^P)}. \quad (14)$$

The sentence with the lowest repair score will be eliminated from the summary. If the summary length is still greater than the maximum length constraint, then this operation is repeated until the condition is satisfied.

E. Main steps of the algorithm

The proposed algorithm, the multi-objective number-one-selection genetic algorithm (MONOGA), is based on the nondominated sorting genetic algorithm II (NSGA-II) [31] but uses all the previously explained specifically designed operators. The principal characteristics of NSGA-II are its reduced computational complexity, its elitist approach and its genetic operators, among others. The main steps of MONOGA are detailed in Algorithm 2.

Algorithm 2 MONOGA pseudocode.

Input: pop_{size} : size of the population, $cycles_{max}$: maximum number of cycles or generations.

Output: NDS : Set of nondominated solutions.

```

1:  $NDS \leftarrow \emptyset$ 
2:  $Pop \leftarrow InitializePopulation()$ 
3:  $Pop \leftarrow RankAndCrowding(Pop, pop_{size})$ 
4: for  $cycle = 1$  to  $cycles_{max}$  do
5:   for  $indiv = 1$  to  $pop_{size}$  do
6:      $Parent1, Parent2 \leftarrow N1Selection(Pop, indiv)$ 
7:      $Off \leftarrow PACrossover(Parent1, Parent2)$ 
8:      $Pop[pop_{size} + indiv] \leftarrow PAMutation(Off)$ 
9:   end for
10:   $Pop \leftarrow RankAndCrowding(Pop, 2 \cdot pop_{size})$ 
11:   $NDS \leftarrow NDS \cup SaveNDS(Pop, pop_{size} + 1, 2 \cdot pop_{size})$ 
12: end for
13:  $NDS \leftarrow NDS \cup SaveNDS(Pop, 1, pop_{size})$ 
14: return  $NDS$ 

```

First, a set of nondominated solutions, NDS , is initialized to an empty set in Line 1. Then, in Line 2, the population Pop is initialized: all the individual solutions are randomly generated. After that, the two multi-objective operators (rank and crowding) are applied to the entire population in Line 3. The first one ranks the solutions in different Pareto fronts depending on the dominance relation among them, and the

second one considers the crowding distance among the solutions preferring the most diverse ones [31]. After this step, the population is sorted in descending order of quality according to the rank and crowding values.

The steps from Lines 4 to 12 are repeated during a maximum number of cycles or generations $cycles_{max}$. Now, the three operations from Lines 6, 7 and 8 are performed for every individual solution of the population. They are the selection, crossover and mutation operations. First, the selection operator is carried out, choosing the two parents $Parent1$ and $Parent2$ from the population based on the number-one selection. Second, the problem-aware crossover operator is performed with the selected parents, and the generated offspring solution is stored in Off . Third, the problem-aware mutation operator is applied over the offspring solution, and the resulting solution is stored in its corresponding position. When these three operators are applied over the entire population (parent population), an offspring population is generated, that is, the size of the population is duplicated. After that, in Line 10, the multi-objective operators (rank and crowding) are implemented again to sort the population, whose size is now $2 \cdot pop_{size}$. In this way, its first half (the one with the best solutions) will be the parent population of the next generation (cycle). At the end of the main loop, in Line 11, the nondominated solutions in the second half of the population are stored in the NDS set to save them.

Finally, in Line 13, the nondominated solutions in the first half of the population are also stored in the NDS set. Figure 1 shows a flowchart for MONOGA.

F. Reduction of the Pareto front to one solution (summary)

The result obtained by MONOGA is a set of nondominated solutions or Pareto front. Any one of the solutions in this set may be chosen as a single final solution. Nevertheless, this choice should be made by using an objective method. A study of some methods for the reduction of the Pareto front to a single solution was performed.

In this paper, four main methods have been developed and tested. These methods are based on the following: the hypervolume, the consensus solution, the shortest distance to the ideal point, and the shortest distance to all points. For the methods based on distances, the shortest distance to the ideal point has analyzed the Euclidean, Manhattan, Chebyshev, and Mahalanobis distances, whereas the shortest distance to all points has analyzed these same distances in addition to the Levenshtein distance. Thus, a total of 11 methods were considered. The formulation and details of these methods can be found in [32].

V. EXPERIMENTAL RESULTS

This section includes the following: the datasets, the evaluation metrics, the experimental settings, the results obtained, the contribution to the result quality of all the MONOGA operators, the comparison with other approaches, and the multi-objective evaluation.

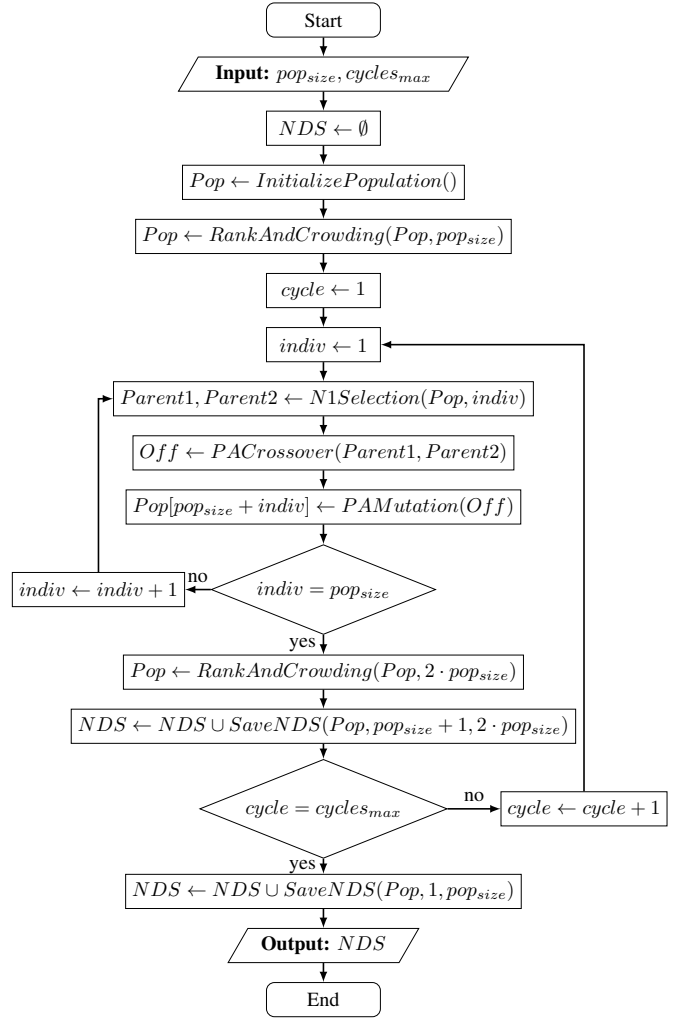


Fig. 1. MONOGA flowchart.

A. Datasets

The datasets used were supplied by the Text Analysis Conference (TAC). The TAC datasets are open benchmarks for automatic summarization evaluation from the National Institute of Standards and Technology (NIST, USA). Specifically, the datasets of the Update Summarization Task from TAC2008 [11] and TAC2009 [12] have been considered. The goal of this task is to produce a multidocument summary about news articles assuming that the user has already read a set of earlier articles to inform him or her about the new information concerning a particular topic. Table I reports some characteristics of the Update Summarization Task datasets from TAC.

B. Preprocessing

Each one of the document collections from the TAC2008 and TAC2009 datasets must be preprocessed prior to the execution of the algorithm, so the following steps must be conducted:

- 1) Segmentation. The sentences from the document collection are extracted in a separate way to determine their beginning and ending.

TABLE I
CHARACTERISTICS OF THE UPDATE SUMMARIZATION TASK DATASETS
FROM TAC2008 AND TAC2009.

Description	TAC2008	TAC2009
Number of topics	48	44
Number of collections per topic	2 (A, B)	2 (A, B)
Number of documents	10, 10	10, 10
Average number of sentences	148, 138	154, 138
Average number of total terms	5129, 4846	5513, 4961
Average number of different terms	628, 640	662, 635
Summary length constraint	100	100

- 2) Tokenization. The words from every sentence are isolated, taken into account as a token, as are the blank spaces. In this step, the punctuation, exclamation, interrogation and other common marks are removed.
- 3) Stop word removal. The words with no meaning are removed from each sentence, such as the articles, prepositions, conjunctions and others. The list of stop words has been supplied by the ROUGE package [33].
- 4) Stemming. Finally, the roots of the remaining words are pulled out using the Porter stemming algorithm [34]. In this way, the words with the same lexical root will be taken into account as the same term.

In addition, only the sentences that are relevant to the query will be used by the algorithm. Specifically, the sentences that do not have any terms in common with the query will be discarded.

C. Evaluation metrics

The metrics considered to evaluate the quality of the generated summaries have been the Recall-Oriented Understudy for Gisting Evaluation (ROUGE) metrics [13]. The ROUGE metrics are commonly used for automatic summarization. They measure the similarity between a system-generated summary and a human-generated summary by calculating the total number of overlapping units.

The ROUGE scores that have been considered are: ROUGE- N , ROUGE-L and ROUGE-SU4. First, ROUGE- N is the N -gram recall between a system-generated summary and a set of human-generated summaries. Two ROUGE- N scores were analyzed, ROUGE-1 and ROUGE-2. Second, ROUGE-L measures the relation between the length of the longest common subsequence of the system-generated summary, and the length of the human-generated summary. Finally, ROUGE-SU4 is considered. It is the skip-bigram co-occurrence (with a maximum length of 4) that computes the overlap of the skip bigrams between a system-generated summary and a set of human-generated summaries.

D. Experimental settings

The experimental settings of MONOGA contain the following parameters, the population size, pop_{size} , the maximum number of generations or cycles, $cycles_{max}$, the mutation probability, p_m and the crossover probability, p_c . The values established for these parameters have been, $pop_{size} = 64$, $cycles_{max} = 500$, $p_m = 1/n$, and $p_c = 100\%$.

For the method chosen for the reduction of the Pareto front, the result of the study described in Subsection IV-F reported that the consensus solution method achieved the best results in terms of ROUGE scores. This method outperformed the other methods both in the mean and number of topics; therefore, it is the one used in all the experiments.

With the aim of providing reliable results, the experimental results presented are the outcomes from a total of 31 repetitions or independent runs per experiment. The experiments have been carried out in a compute node consisting of 4 AMD Opteron Abu Dhabi 6376 2.3 GHz processors, with 96-GB RAM. The approach has been developed in the C/C++ language in the Eclipse Platform on Ubuntu 20.04.2.0 LTS (Focal Fossa).

E. Comparison with the Nondominated Sorting Genetic Algorithm (NSGA-II)

In this subsection, the results obtained with MONOGA in the TAC2008 and TAC2009 datasets are presented. For each of the evaluated ROUGE scores, the results reported are the mean for all the topics from each dataset and the standard deviation (Std. dev). In addition, a comparison with the standard multi-objective evolutionary algorithm NSGA-II is also performed here. To make fair comparisons, NSGA-II has been configured with the same parameter settings as MONOGA. Table II shows the results provided by MONOGA and by NSGA-II.

TABLE II
COMPARISON OF THE RESULTS OBTAINED BY MONOGA AND BY
NSGA-II IN THE TAC2008 AND TAC2009 DATASETS. THE BEST MEAN
VALUES ARE HIGHLIGHTED IN BOLD.

		ROUGE-1	ROUGE-2	ROUGE-L	ROUGE-SU4
TAC2008					
MONOGA	Mean	0.42761	0.11563	0.36146	0.15856
	Std. dev	0.06246	0.05164	0.05534	0.03765
NSGA-II	Mean	0.40447	0.09522	0.33999	0.14256
	Std. dev	0.05570	0.04883	0.05019	0.03256
TAC2009					
MONOGA	Mean	0.42163	0.11471	0.34476	0.15477
	Std. dev	0.04887	0.05380	0.04472	0.03446
NSGA-II	Mean	0.39975	0.09600	0.32644	0.14011
	Std. dev	0.04964	0.04640	0.04251	0.03151

As shown in Table II, MONOGA outperforms NSGA-II in the four ROUGE scores, both in the TAC2008 and TAC2009 datasets. To analyze the enhancements achieved, Table III reports the percentage improvements.

TABLE III
PERCENTAGE IMPROVEMENTS OBTAINED BY MONOGA WITH RESPECT
TO NSGA-II IN THE TAC2008 AND TAC2009 DATASETS.

	ROUGE-1	ROUGE-2	ROUGE-L	ROUGE-SU4	Average
TAC2008	5.72%	21.44%	6.32%	11.23%	11.18%
TAC2009	5.47%	19.49%	5.61%	10.46%	10.26%

The percentages in Table III show that the average improvement is over 10% in both datasets, with a range of values from 5.47% to 21.44%. Specifically, the percentage improvements are approximately 6% in the ROUGE-1 and ROUGE-L scores,

11% for the ROUGE-SU4 scores, and 20% for the ROUGE-2 scores.

Now, the contribution of each MONOGA operator to the result quality is analyzed in the following subsections. The order followed has been from lowest to highest contribution, so the crossover operator will be studied first, then the mutation operator, and finally the selection operator.

F. Contribution of the problem-aware crossover operator

For MONOGA, this operator has been designed in a problem-aware way, while NSGA-II uses single-point crossover. A version of MONOGA with single-point crossover has been developed to evaluate its performance, named MONOGA-SPC. Table IV shows the results obtained by MONOGA and MONOGA-SPC in the TAC2008 and TAC2009 datasets.

TABLE IV

COMPARISON OF THE RESULTS OBTAINED BY MONOGA AND BY MONOGA-SPC IN THE TAC2008 AND TAC2009 DATASETS. THE BEST MEAN VALUES ARE HIGHLIGHTED IN BOLD.

		ROUGE-1	ROUGE-2	ROUGE-L	ROUGE-SU4
TAC2008					
MONOGA	Mean	0.42761	0.11563	0.36146	0.15856
	Std. dev	0.06246	0.05164	0.05534	0.03765
MONOGA-SPC	Mean	0.41512	0.10327	0.35221	0.15108
	Std. dev	0.06394	0.05135	0.05738	0.03841
TAC2009					
MONOGA	Mean	0.42163	0.11471	0.34476	0.15477
	Std. dev	0.04887	0.05380	0.04472	0.03446
MONOGA-SPC	Mean	0.40886	0.10406	0.33674	0.14873
	Std. dev	0.04731	0.05075	0.04194	0.03323

The results in Table IV report that MONOGA outperforms MONOGA-SPC in all the ROUGE scores in both datasets. Now, Table V presents the percentage improvements attained by MONOGA.

TABLE V

PERCENTAGE IMPROVEMENTS OBTAINED BY MONOGA WITH RESPECT TO MONOGA-SPC IN THE TAC2008 AND TAC2009 DATASETS.

	ROUGE-1	ROUGE-2	ROUGE-L	ROUGE-SU4	Average
TAC2008	3.01%	11.98%	2.63%	4.95%	5.64%
TAC2009	3.12%	10.24%	2.38%	4.06%	4.95%

The percentage improvements shown in Table V report that the average improvement is approximately 5%. In addition, these values are between 2.38% and 11.98%. The ROUGE-L scores achieve improvements over 2%, the ROUGE-1 scores approximately 3%, the ROUGE-SU4 scores over 4% and the ROUGE-2 scores approximately 11%. Hence, the use of the problem-aware crossover operator rather than the single-point crossover operator causes relevant performance improvements for the proposed MONOGA.

G. Contribution of the problem-aware mutation operator

The mutation operator carried out in MONOGA has been problem-aware designed, whereas NSGA-II uses a bitwise mutation operator. To study the performance of this operator, a version of MONOGA with a bitwise mutation, called

MONOGA-BWM, has been developed. Table VI provides information on the results obtained by MONOGA and by MONOGA-BWM in the TAC2008 and TAC2009 datasets.

TABLE VI

COMPARISON OF THE RESULTS OBTAINED BY MONOGA AND BY MONOGA-BWM IN THE TAC2008 AND TAC2009 DATASETS. THE BEST MEAN VALUES ARE HIGHLIGHTED IN BOLD.

		ROUGE-1	ROUGE-2	ROUGE-L	ROUGE-SU4
TAC2008					
MONOGA	Mean	0.42761	0.11563	0.36146	0.15856
	Std. dev	0.06246	0.05164	0.05534	0.03765
MONOGA-BWM	Mean	0.40833	0.09947	0.34984	0.14913
	Std. dev	0.06237	0.04681	0.05575	0.03863
TAC2009					
MONOGA	Mean	0.42163	0.11471	0.34476	0.15477
	Std. dev	0.04887	0.05380	0.04472	0.03446
MONOGA-BWM	Mean	0.40394	0.09997	0.33443	0.14564
	Std. dev	0.05330	0.05162	0.04820	0.03756

As shown in Table VI, the results obtained by MONOGA improve the ones obtained by MONOGA-BWM. These improvements take place in the four ROUGE scores both in the TAC2008 and TAC2009 datasets. Table VII presents the percentage improvements provided by MONOGA to study the difference in performance between the mutation operators.

TABLE VII

PERCENTAGE IMPROVEMENTS OBTAINED BY MONOGA WITH RESPECT TO MONOGA-BWM IN THE TAC2008 AND TAC2009 DATASETS.

	ROUGE-1	ROUGE-2	ROUGE-L	ROUGE-SU4	Average
TAC2008	4.72%	16.24%	3.32%	6.33%	7.65%
TAC2009	4.38%	14.75%	3.09%	6.27%	7.12%

The values reported in Table VII show that the average improvement is approximately 7%, and the percentages range between 3.09% and 16.24%. Particularly, for the ROUGE-L scores the improvements are over 3%, for the ROUGE-1 scores over 4%, for the ROUGE-SU4 scores over 6% and for the ROUGE-2 scores approximately 15%. It can be concluded that the use of the problem-aware mutation operator, instead of the bitwise mutation operator, produces relevant improvements.

H. Contribution of the number-one selection operator

The selection operator in MONOGA, the number-one selection, has been proposed in this work. Its performance is compared with the binary tournament selection used by NSGA-II. A version of MONOGA with binary tournament selection, MONOGA-BTS, has been developed. Table VIII shows the results obtained by MONOGA and MONOGA-BTS on the TAC2008 and TAC2009 datasets. Table VIII shows that MONOGA outperforms MONOGA-BTS. The improvements occur in all the ROUGE scores in both the TAC2008 and TAC2009 datasets.

Table IX shows the percentage improvements achieved by MONOGA regarding MONOGA-BTS with the aim of highlighting the impact of using the number-one selection operator instead of the traditional operator. As can be observed, the average improvement is 10.01% and 9.07% for the TAC2008 and TAC2009 datasets, respectively. The percentages range

TABLE VIII
COMPARISON OF THE RESULTS OBTAINED BY MONOGA AND BY MONOGA-BTS IN THE TAC2008 AND TAC2009 DATASETS. THE BEST MEAN VALUES ARE HIGHLIGHTED IN BOLD.

		ROUGE-1	ROUGE-2	ROUGE-L	ROUGE-SU4
TAC2008					
MONOGA	Mean	0.42761	0.11563	0.36146	0.15856
	Std. dev	0.06246	0.05164	0.05534	0.03765
MONOGA-BTS	Mean	0.40511	0.09599	0.34424	0.14543
	Std. dev	0.05343	0.04575	0.04806	0.03248
TAC2009					
MONOGA	Mean	0.42163	0.11471	0.34476	0.15477
	Std. dev	0.04887	0.05380	0.04472	0.03446
MONOGA-BTS	Mean	0.40171	0.09178	0.33011	0.14216
	Std. dev	0.04689	0.04742	0.04070	0.03063

from 4.44% to 20.46%, and is approximately 5% in the ROUGE-1 and ROUGE-L scores, and is approximately 9% for the ROUGE-SU4 scores, and is approximately 19% in the ROUGE-2 scores. Therefore, the use of the number-one selection operator instead of the binary tournament selection operator produces very relevant performance improvements.

TABLE IX
PERCENTAGE IMPROVEMENTS OBTAINED BY MONOGA WITH RESPECT TO MONOGA-BTS IN THE TAC2008 AND TAC2009 DATASETS.

	ROUGE-1	ROUGE-2	ROUGE-L	ROUGE-SU4	Average
TAC2008	5.55%	20.46%	5.00%	9.03%	10.01%
TAC2009	4.96%	18.03%	4.44%	8.86%	9.07%

I. Comparison with other approaches

The results obtained by MONOGA are compared with those from the scientific literature. The eleven works in Section II present some results that can be used for comparisons. However, they only show the ROUGE means but not the standard deviations. Table X presents the comparison of the results obtained by MONOGA and those obtained by the other approaches that considered the TAC2008 or TAC2009 datasets, showing the symbol “-” when the corresponding ROUGE score is not available.

As seen in Table X, MONOGA outperforms all the other approaches that considered the TAC2008 or TAC2009 datasets. Table XI shows the percentage improvements produced by MONOGA. This table shows that, on average, the performance of MONOGA is superior, showing improvements between 12.74% and 21.41% in the TAC2008 datasets, and between 14.35% and 55.03% in the TAC2009 datasets. For the TAC2008 datasets, taking into account the four ROUGE scores, the average percentage improvement obtained is 16.98%. In the same way, considering the four ROUGE scores, the average percentage improvement reached for the TAC2009 datasets is 26.44%. In conclusion, the proposed MONOGA clearly improves on the performance of the existing approaches in the scientific literature.

J. Multi-objective evaluation

In this subsection, the proposed MONOGA is evaluated from the multi-objective optimization point of view. For comparison purposes, three widely used multi-objective evolutionary

TABLE X
COMPARISON OF THE RESULTS OBTAINED BY MONOGA AND BY OTHER APPROACHES IN THE TAC2008 AND TAC2009 DATASETS. THE BEST VALUES ARE HIGHLIGHTED IN BOLD.

Approach	ROUGE-1	ROUGE-2	ROUGE-L	ROUGE-SU4
TAC2008				
MONOGA	0.42761	0.11563	0.36146	0.15856
KLear [26]	-	0.10400	-	-
Game Theory [25]	-	0.09489	-	0.13467
IMDDS [24]	-	0.10100	-	0.13700
MCL-ILP [23]	0.38620	0.11570	-	0.14750
Supervised ILP [22]	-	0.09990	-	0.13610
QCQPSum [20]	0.37002	0.08812	-	-
EEDS [21]	-	0.09773	-	0.13283
MarginRank [18]	0.36096	0.08569	0.31633	0.12850
CBSEAS [19]	-	0.08950	-	0.12950
MRSP [14]	-	0.10217	-	0.13778
Information Distance [15]	0.37600	-	-	-
MWDS [16]	-	0.08117	-	0.11728
IHCUS [17]	0.37100	0.09300	0.32500	0.13500
Mean other approaches	0.37284	0.09607	0.32067	0.13362
TAC2009				
MONOGA	0.42163	0.11471	0.34476	0.15477
KLear [26]	-	0.10300	-	-
MCL-ILP [23]	0.37530	0.10600	-	0.14080
Supervised ILP [22]	-	0.09610	-	0.13770
QCQPSum [20]	0.36247	0.08610	-	-
MarginRank [18]	0.35415	0.08357	0.22239	0.12615
MRSP [14]	-	0.09932	-	0.13771
Information Distance [15]	0.36800	-	-	-
Mean other approaches	0.36498	0.09568	0.22239	0.13559

TABLE XI
PERCENTAGE IMPROVEMENTS OF MONOGA WITH RESPECT TO OTHER APPROACHES IN THE TAC2008 AND TAC2009 DATASETS.

Approach	ROUGE-1	ROUGE-2	ROUGE-L	ROUGE-SU4
TAC2008				
KLear [26]	-	11.19%	-	-
Game Theory [25]	-	21.86%	-	17.74%
IMDDS [24]	-	14.49%	-	15.74%
MCL-ILP [23]	10.72%	-0.06%	-	7.50%
Supervised ILP [22]	-	15.75%	-	16.50%
QCQPSum [20]	15.56%	31.22%	-	-
EEDS [21]	-	18.32%	-	19.37%
MarginRank [18]	18.46%	34.94%	14.27%	23.39%
CBSEAS [19]	-	13.18%	-	15.08%
MRSP [14]	-	29.20%	-	22.44%
Information Distance [15]	13.73%	-	-	-
MWDS [16]	-	42.46%	-	35.20%
IHCUS [17]	15.26%	24.34%	11.22%	17.45%
Mean other approaches	14.75%	21.41%	12.74%	19.04%
TAC2009				
KLear [26]	-	11.37%	-	-
MCL-ILP [23]	12.34%	8.22%	-	9.92%
Supervised ILP [22]	-	19.37%	-	12.39%
QCQPSum [20]	16.32%	33.23%	-	-
MarginRank [18]	19.05%	37.26%	55.03%	22.68%
MRSP [14]	-	15.50%	-	12.39%
Information Distance [15]	14.57%	-	-	-
Mean other approaches	15.57%	20.82%	55.03%	14.35%

algorithms have been adapted for the update summarization problem and implemented, specifically, NSGA-II, the strength Pareto evolutionary algorithm 2 (SPEA2) and the fast Pareto genetic algorithm (FastPGA). On the one hand, SPEA2 is an improved version of the basic SPEA [35]. The basic SPEA seeks to find the Pareto-optimal set by combining several features. It keeps a set of individuals that contains the Pareto-optimal solutions generated thus far; this set is used to assess the fitness value of every individual (depending on the

Pareto dominance); population diversity is maintained without requiring any distance parameter; and a clustering method is performed to reduce the size of the Pareto set. Specifically, SPEA2 incorporates a fine-grained fitness assignment strategy, a density estimation method and an improved archive truncation technique. FastPGA uses a particular fitness assignment and ranking strategy for the simultaneous optimization of multiple objectives [36]. To ensure the fast propagation of the Pareto optimal solution set, an elitism operator is specifically developed. In addition, a population regulation operator is used to dynamically adapt the population size as needed.

The multi-objective evaluation in this paper consists of measuring the performance of the four algorithms (MONOGA, NSGA-II, SPEA2 and FastPGA) by using two metrics, the hypervolume (HV) and the inverted generational distance (IGD). HV is a widely used performance indicator in the field of multi-objective optimization [37]. It measures the portion or percentage of the objective space that is covered by a set of nondominated solutions (Pareto front). Since the presented problem involves the maximization of three objective functions, the portion is the covered volume, and the larger the HV is, the better the performance. IGD is also a commonly used metric in the multi-objective field, which computes the mean distance between every solution in the assessed Pareto front and in the optimal one [38]. Hence, a small IGD value means that the assessed Pareto front is very close to the optimal one.

Table XII shows the results obtained by the four algorithms for the HV metric in the TAC2008 and TAC2009 datasets. MONOGA has provided the best mean HV values in both datasets (approximately 68%). Regarding the other algorithms, the mean HV values are similar (approximately 60%).

TABLE XII
RESULTS OBTAINED BY MONOGA, NSGA-II, SPEA2 AND FASTPGA FOR THE HV METRIC IN THE TAC2008 AND TAC2009 DATASETS. THE BEST VALUES ARE HIGHLIGHTED IN BOLD.

	MONOGA	NSGA-II	SPEA2	FastPGA
TAC2008				
Mean	68.39%	60.91%	59.23%	61.14%
Median	68.56%	59.90%	58.68%	60.23%
Standard deviation	6.22%	8.34%	9.01%	8.38%
Q_1	63.82%	55.16%	52.50%	55.47%
Q_3	72.87%	65.36%	64.49%	65.63%
TAC2009				
Mean	67.30%	60.34%	59.03%	60.60%
Median	66.09%	60.07%	58.79%	60.35%
Standard deviation	7.24%	8.69%	8.85%	8.72%
Q_1	61.76%	53.52%	52.67%	53.66%
Q_3	72.01%	66.35%	65.89%	66.77%

Table XIII presents the results obtained by the four algorithms for the IGD metric in the TAC2008 and TAC2009 datasets. This table reports that MONOGA achieved the best mean IGD values in both datasets (approximately 1.10). For the other algorithms, the obtained IGD values were very similar (approximately 1.24).

To provide visual statistical results, Figure 2 shows the boxplots obtained by the four algorithms for the HV and

TABLE XIII
RESULTS OBTAINED BY MONOGA, NSGA-II, SPEA2 AND FASTPGA FOR THE IGD METRIC IN THE TAC2008 AND TAC2009 DATASETS. THE BEST VALUES ARE HIGHLIGHTED IN BOLD.

	MONOGA	NSGA-II	SPEA2	FastPGA
TAC2008				
Mean	1.10397	1.24948	1.25185	1.24934
Median	1.10125	1.25196	1.28592	1.25196
Standard deviation	0.05744	0.07061	0.07869	0.07070
Q_1	1.06911	1.22715	1.23162	1.22672
Q_3	1.13414	1.28280	1.29632	1.28242
TAC2009				
Mean	1.10152	1.23852	1.23808	1.23839
Median	1.10529	1.24709	1.25510	1.24661
Standard deviation	0.05667	0.08280	0.09076	0.08281
Q_1	1.07401	1.21362	1.21997	1.21363
Q_3	1.14439	1.28451	1.29349	1.28430

IGD metrics in the TAC2008 and TAC2009 datasets. In addition, a hypothesis test-based statistical study was carried out. Statistically significant differences were searched, and two-sided p -values smaller than 0.05 were considered statistically significant. ANOVA tests were used when the applicability conditions could be assumed (normality and homoscedasticity), otherwise, the Kruskal-Wallis tests were applied. In the significant cases, Bonferroni or nonparametric post hoc pairwise comparisons were applied. Tables XIV and XV present the statistically significant differences obtained from pairwise comparisons for the four algorithms in the TAC2008 and TAC2009 datasets for the HV and IGD metrics, respectively. From the results reported in Tables XIV and XV, it can be concluded that MONOGA provides significantly better values (HV and IGD) than the corresponding ones from NSGA-II, SPEA2 and FastPGA. Therefore, the multi-objective performance of MONOGA is better than the other algorithms.

TABLE XIV
 p -VALUES FROM A PAIRWISE COMPARISON BETWEEN MONOGA AND THE OTHER ALGORITHMS FOR THE HV METRIC IN THE TAC2008 AND TAC2009 DATASETS. THE STATISTICALLY SIGNIFICANT DIFFERENCES ARE HIGHLIGHTED IN BOLD.

	MONOGA	NSGA-II	SPEA2	FastPGA
TAC2008				
MONOGA	-	<0.001	<0.001	<0.001
NSGA-II		-	1.000	1.000
SPEA2			-	1.000
FastPGA				-
TAC2009				
MONOGA	-	<0.001	<0.001	<0.001
NSGA-II		-	1.000	1.000
SPEA2			-	1.000
FastPGA				-

Finally, Table XVI presents the average execution times of the four algorithms in the TAC2008 and TAC2009 datasets. As can be observed, NSGA-II has the smallest execution times. This is due to the simplicity of its generic basic operators. MONOGA and FastPGA have provided similar times. Even

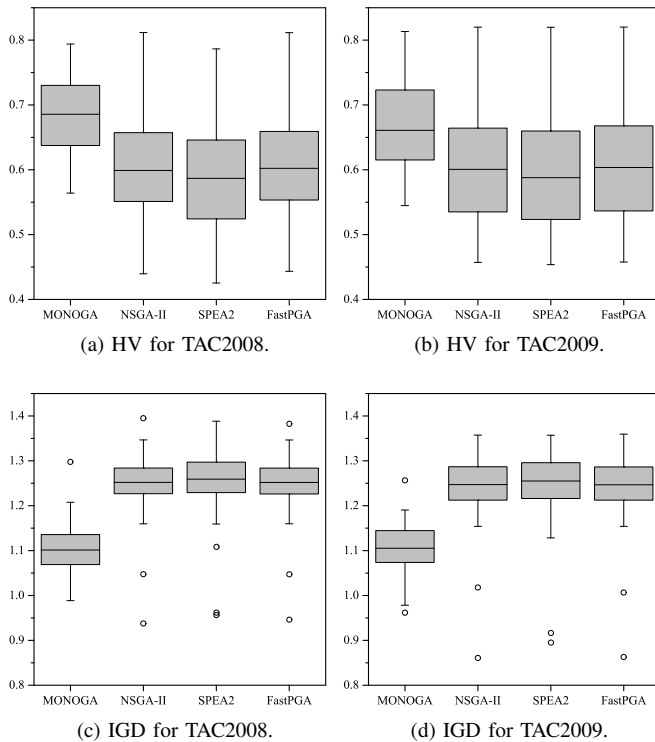


Fig. 2. Boxplots for HV and IGD metrics in the TAC2008 and TAC2009 datasets.

TABLE XV

p-VALUES FROM A PAIRWISE COMPARISON BETWEEN MONOGA AND OTHER ALGORITHMS FOR THE IGD METRIC IN THE TAC2008 AND TAC2009 DATASETS. THE STATISTICALLY SIGNIFICANT DIFFERENCES ARE HIGHLIGHTED IN BOLD.

	MONOGA	NSGA-II	SPEA2	FastPGA
TAC2008				
MONOGA	-	< 0.001	< 0.001	< 0.001
NSGA-II		-	0.690	0.979
SPEA2			-	0.671
FastPGA				-
TAC2009				
MONOGA	-	< 0.001	< 0.001	< 0.001
NSGA-II		-	0.833	0.977
SPEA2			-	0.811
FastPGA				-

with specifically problem-aware designed operators, the execution times of MONOGA have been kept low. Finally, SPEA2 provides the largest execution times.

TABLE XVI

AVERAGE EXECUTION TIMES (IN SECONDS) FOR EACH ALGORITHM IN THE TAC2008 AND TAC2009 DATASETS. THE BEST VALUES IN BOLD.

	MONOGA	NSGA-II	SPEA2	FastPGA
TAC2008	5.194	2.813	12.855	4.979
TAC2009	6.090	3.462	13.594	5.880

VI. CONCLUSION

Update summarization is one of the most challenging automatic summarization problems, since it involves the handling of dynamic document collections. In the scientific literature, there is a lack of multi-objective approaches that address this problem. For this reason, the Multi-Objective Number-One-selection Genetic Algorithm (MONOGA) has been designed and implemented. The main operators, number-one selection, problem-aware crossover and problem-aware mutation, have been specifically designed for this task. The results obtained by MONOGA have improved, on average, the ones obtained by NSGA-II by more than 10% in the ROUGE scores. The operators that have contributed the most to this improvement have been the number-one selection operator, followed by the problem-aware mutation and problem-aware crossover operators. In addition, MONOGA has also outperformed the results of the existing approaches in the scientific literature, with average percentage improvements between 12.74% and 55.03% in the ROUGE scores. For the multi-objective evaluation, MONOGA has achieved the best performance with the Hypervolume (HV) and Inverse Generational Distance (IGD) indicators.

In future work, the update summarization problem will be addressed in conjunction with sentiment analysis. In this way, the sentiment-oriented update summarization problem will generate sentiment-oriented update summaries, which will take into account both the relevant new information delivered to the user and the sentiment orientation of the sentences included in it. Furthermore, because of the good results of the number-one selection operator, it will be used in other evolutionary algorithms and applications.

ACKNOWLEDGMENTS

This research was supported by the Ministry of Science, Innovation and Universities - Spain and State Research Agency - Spain (Projects PID2019-107299GB-I00 and PID2021-122209OB-C32 funded by MCIN/AEI/10.13039/501100011033), Junta de Extremadura - Spain (Projects GR21017 and GR21057), and European Union (European Regional Development Fund). Jesus M. Sanchez-Gomez is 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, and M. R. Yeganegi, "Text mining in big data analytics," *Big Data and Cognitive Computing*, vol. 4, no. 1, p. 1, 2020.
- [2] W. S. El-Kassas, C. R. Salama, A. A. Rafea, and H. K. Mohamed, "Automatic text summarization: A comprehensive survey," *Expert Systems with Applications*, vol. 165, p. 113679, 2021.
- [3] J.-Y. Delort and E. Alfonseca, "DualSum: a topic-model based approach for update summarization," in *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, 2012, pp. 214–223.
- [4] M. Afsharizadeh, H. Ebrahimipour-Komleh, and A. Bagheri, "Query-oriented text summarization using sentence extraction technique," in *2018 4th International Conference on Web Research (ICWR)*. IEEE, 2018, pp. 128–132.
- [5] R. M. Alguliyev, R. M. Aliguliyev, N. R. Isazade, A. Abdi, and N. Idris, "COSUM: Text summarization based on clustering and optimization," *Expert Systems*, vol. 36, no. 1, p. e12340, 2019.

- [6] L. Huang, W. Chen, Y. Liu, S. Hou, and H. Qu, "Summarization with self-aware context selecting mechanism," *IEEE Transactions on Cybernetics*, vol. 52, no. 7, pp. 5828–5841, 2022.
- [7] K. Yao, L. Zhang, D. Du, T. Luo, L. Tao, and Y. Wu, "Dual encoding for abstractive text summarization," *IEEE Transactions on Cybernetics*, vol. 50, no. 3, pp. 985–996, 2020.
- [8] Y. Zhang, M. J. Er, R. Zhao, and M. Pratama, "Multiview Convolutional Neural Networks for Multidocument Extractive Summarization," *IEEE Transactions on Cybernetics*, vol. 47, no. 10, pp. 3230–3242, 2017.
- [9] R. Rautray and R. C. Balabantaray, "An evolutionary framework for multi document summarization using Cuckoo search approach: MD-SCSA," *Applied Computing and Informatics*, vol. 14, no. 2, pp. 134–144, 2018.
- [10] J. M. Sanchez-Gomez, M. A. Vega-Rodríguez, and C. J. Pérez, "Extractive multi-document text summarization using a multi-objective artificial bee colony optimization approach," *Knowledge-Based Systems*, vol. 159, pp. 1–8, 2018.
- [11] H. T. Dang and K. Owczarzak, "Overview of the TAC 2008 Update Summarization Task," in *Proceedings of the Text Analysis Conference (TAC 2008)*. National Institute of Standards and Technology, 2008, pp. 1–16.
- [12] —, "Overview of the TAC 2009 Summarization Track," in *Text Analysis Conference (TAC)*. National Institute of Standards and Technology, 2009, pp. 111–113.
- [13] 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.
- [14] P. Du, J. Guo, J. Zhang, and X. Cheng, "Manifold ranking with sink points for update summarization," in *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*. Association for Computing Machinery, 2010, pp. 1757–1760.
- [15] C. Long, M.-L. Huang, X.-Y. Zhu, and M. Li, "A New Approach for Multi-Document Update Summarization," *Journal of Computer Science and Technology*, vol. 25, no. 4, pp. 739–749, 2010.
- [16] C. Shen and T. Li, "Multi-document summarization via the minimum dominating set," in *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*. Association for Computational Linguistics, 2010, pp. 984–992.
- [17] D. Wang and T. Li, "Document update summarization using incremental hierarchical clustering," in *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*. Association for Computing Machinery, 2010, pp. 279–288.
- [18] X. Li, L. Du, and Y.-D. Shen, "Graph-Based Marginal Ranking for Update Summarization," in *Proceedings of the 2011 SIAM International Conference on Data Mining*. Society for Industrial and Applied Mathematics, 2011, pp. 486–497.
- [19] A. Bossard and C. Rodrigues, "Combining a Multi-Document Update Summarization System –CBSEAS– with a Genetic Algorithm," in *Combinations of Intelligent Methods and Applications*. Springer, 2011, pp. 71–87.
- [20] X. Li, L. Du, and Y.-D. Shen, "Update summarization via graph-based sentence ranking," *IEEE Transactions on Knowledge and Data Engineering*, vol. 25, no. 5, pp. 1162–1174, 2012.
- [21] R. He, B. Qin, and T. Liu, "A novel approach to update summarization using evolutionary manifold-ranking and spectral clustering," *Expert Systems with Applications*, vol. 39, no. 3, pp. 2375–2384, 2012.
- [22] C. Li, Y. Liu, and L. Zhao, "Improving Update Summarization via Supervised ILP and Sentence Reranking," in *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, 2015, pp. 1317–1322.
- [23] M. Mnasri, G. de Chalendar, and O. Ferret, "Taking into account Inter-sentence Similarity for Update Summarization," in *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*. Association for Computational Linguistics, 2017, pp. 204–209.
- [24] M. Liu, W. Wei, B. Xu, and H. Wang, "Study on Dynamic Multi-document Summarization System Framework Method," *Journal of Internet Technology*, vol. 19, no. 5, pp. 1339–1346, 2018.
- [25] A. Ahmad and T. Ahmad, "A Game Theory Approach for Multi-document Summarization," *Arabian Journal for Science and Engineering*, vol. 44, no. 4, pp. 3655–3667, 2019.
- [26] M. Peyrard and R. West, "KLearn: Background Knowledge Inference from Summarization Data," in *Findings of the Association for Computational Linguistics: EMNLP 2020*. Association for Computational Linguistics, 2020, pp. 2073–2085.
- [27] G. Salton and C. Buckley, "Term-weighting approaches in automatic text retrieval," *Information Processing & Management*, vol. 24, no. 5, pp. 513–523, 1988.
- [28] R. M. Alguliev, R. M. Aliguliyev, and M. S. Hajirahimova, "Gen-DocSum+MCLR: Generic document summarization based on maximum coverage and less redundancy," *Expert Systems with Applications*, vol. 39, no. 16, pp. 12460–12473, 2012.
- [29] R. M. Alguliev, R. M. Aliguliyev, and N. R. Isazade, "Multiple documents summarization based on evolutionary optimization algorithm," *Expert Systems with Applications*, vol. 40, no. 5, pp. 1675–1689, 2013.
- [30] R. M. Alguliev, R. M. Aliguliyev, and C. A. Mehdiyev, "Sentence selection for generic document summarization using an adaptive differential evolution algorithm," *Swarm and Evolutionary Computation*, vol. 1, no. 4, pp. 213–222, 2011.
- [31] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.
- [32] J. M. Sanchez-Gomez, M. A. Vega-Rodríguez, and C. J. Pérez, "Comparison of automatic methods for reducing the Pareto front to a single solution applied to multi-document text summarization," *Knowledge-Based Systems*, vol. 174, pp. 123–136, 2019.
- [33] J. Li, "ROUGE Metric," 2020, last accessed: 2-November-2022. [Online]. Available: <https://pypi.org/project/rouge-metric/>
- [34] M. Porter, "The Porter Stemming Algorithm," 2021, last accessed: 2-November-2022. [Online]. Available: <http://www.tartarus.org/martin/PorterStemmer/>
- [35] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the Strength Pareto Evolutionary Algorithm," *TIK-report*, vol. 103, 2001.
- [36] H. Eskandari, C. D. Geiger, and G. B. Lamont, "FastPGA: A Dynamic Population Sizing Approach for Solving Expensive Multiobjective Optimization Problems," in *International Conference on Evolutionary Multi-Criterion Optimization (EMO)*. Springer, 2007, pp. 141–155.
- [37] K. Shang, H. Ishibuchi, L. He, and L. M. Pang, "A Survey on the Hypervolume Indicator in Evolutionary Multiobjective Optimization," *IEEE Transactions on Evolutionary Computation*, vol. 25, no. 1, pp. 1–20, 2020.
- [38] L. C. Bezerra, M. López-Ibáñez, and T. Stützle, "An Empirical Assessment of the Properties of Inverted Generational Distance on Multi- and Many-Objective Optimization," in *International Conference on Evolutionary Multi-Criterion Optimization*. Springer, 2017, pp. 31–45.



Jesus M. Sanchez-Gomez received the Master's degree in Informatics Engineering from the University of Extremadura, Spain, in 2018. He is conducting research in the Department of Computer and Communications Technologies, University of Extremadura, exploring the multi-document text summarization field from different approaches. His main research interests include parallel and distributed computing, evolutionary computation, and multi-objective optimization.



Miguel A. Vega-Rodríguez received the Ph.D. degree in Computer Engineering from the University of Extremadura, Spain, in 2003. He is a Full Professor (Catedrático de Universidad) of computer architecture in the Department of Computer and Communications Technologies, University of Extremadura. He has authored or co-authored more than 740 publications including journal papers (over 180 JCR-indexed journal papers), book chapters, and peer-reviewed conference proceedings, for which he got several awards. He has contributed to the organization of more than 20 international conferences and workshops, namely as general chair or co-chair. He has edited over 20 special issues of JCR-indexed journals. His main research interests include parallel and distributed computing, multiobjective optimization, evolutionary and bio-inspired computation, bioinformatics, and reconfigurable and embedded computing.

ization of more than 20 international conferences and workshops, namely as general chair or co-chair. He has edited over 20 special issues of JCR-indexed journals. His main research interests include parallel and distributed computing, multiobjective optimization, evolutionary and bio-inspired computation, bioinformatics, and reconfigurable and embedded computing.



Carlos J. Pérez received his Master's degree in Mathematical Science in 1996 and his Ph.D. degree in Mathematics in 2003. He is a Full Professor of Statistics in the Department of Mathematics at the University of Extremadura, Spain. He has authored or co-authored more than 75 JCR-indexed journal papers about statistical methodology and applications in diverse knowledge fields, including natural language processing. He has participated in many research projects from competitive calls and contracts. He also has been reviewer for some journals

as Expert Systems with Applications, Reliability Engineering and Safety Systems, Journal of Applied Statistics or Annals of Applied Statistics. His main research interests are in the area of Bayesian statistical inference and classification.