Knowledge-Based Systems 225 (2021) 107113

Contents lists available at ScienceDirect

Knowledge-Based Systems

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

Addressing topic modeling with a multi-objective optimization approach based on swarm intelligence

Carlos González-Santos [a], Miguel A. Vega-Rodríguez [b,*], Carlos J. Pérez [a]

[a] Department of Mathematics, University of Extremadura, Campus Universitario s/n, 10003 Caceres, Spain
[b] Department of Computer and Communications Technologies, University of Extremadura, Campus Universitario s/n, 10003 Caceres, Spain

ARTICLE INFO

Article history:
Received 13 January 2021
Received in revised form 2 April 2021
Accepted 2 May 2021
Available online 3 May 2021

Keywords:
Artificial bee colony
Evolutionary computing
Latent Dirichlet allocation
Multi-objective optimization
Text analysis
Topic modeling

ABSTRACT

Topic modeling is a growing field within the area of text analysis that extracts underlying topics from document collections. Several objectives can be simultaneously considered when designing an approach for topic modeling. A multi-objective optimization approach based on the swarm intelligence of a bee colony (MOABC, Multi-Objective Artificial Bee Colony) has been designed, implemented, and tested. This new approach has been evaluated by using documents from the Reuters-21578 and TagMyNews datasets. Three objective functions (coherence, coverage, and perplexity) and three multi-objective metrics (hypervolume, set coverage, and distance to the ideal point) have been considered in two topic scenarios. Results show that MOABC provides relevant improvements with respect to LDA (Latent Dirichlet Allocation, the most used approach for topic modeling) and MOEA/D (Multi-Objective Evolutionary Algorithm based on Decomposition, the only multi-objective approach published to date). This demonstrates that the multi-criteria nature of topic modeling should be exploited with multi-objective optimization approaches.

1. Introduction

Digital data has been continuously increasing during the last years. Most of the information is stored in document collections such as news, scientific papers, medical reviews, etc. Each one of these collections comprises a huge size, but only some information is important to provide efficient answers to the interested people [1]. This reduced information may cover the main topics of each collection and it must be returned in a reasonable time. Apart from the size, these collections are not organized in a structured database format, but they are organized as unstructured data. In order to process this type of information, it is necessary to use Natural Language Processing (NLP) tools [2]. Text mining provides the techniques required to work with unstructured text data [3]. Some of them are information retrieval [4], information extraction [5], text summarization [6], and topic modeling [7], among others.

Topic Modeling (TM) is a recent branch of natural language processing that is gaining importance in the field of statistical text analysis. It is based on statistical methods that analyze a whole text collection. After a preprocessing of the text collection has been applied, each different word present in the text is taken as a term. The analysis of the group of terms leads to discover topics that represent the collection. For example, an application of TM is the processing of a large document collection containing citations about COVID-19 [8]. Topic models were used to extract a fixed number of topics containing a list of terms that provided a semantic representation of them. So, the most significant terms from each topic described different subjects as the symptoms, the causes of transmission or the treatments for the virus.

In the field of text mining, TM is used for keyword extraction [9], exploratory search [10], document classification [11], sentiment analysis [12], and multilingual analysis [13]. Topic models are classified as supervised or unsupervised [14]. Supervised models only generate those topics that correspond to a label set, whereas unsupervised models generate underlying topics from the terms that form the corpus. Even so, unsupervised models can achieve a totally automated performance with accurate results. The most known unsupervised topic models are Latent Semantic Analysis (LSA) [15] and Latent Dirichlet Allocation (LDA) [16].

On the one hand, LSA is a statistical method that generates a term-document matrix to represent the whole corpus in a latent semantic space. The most relevant terms are selected from this space to compute similarities between terms considering that co-occurrence between terms leads to a semantic similarity [15]. On the other hand, LDA is a generative statistical model that represents documents as mixtures of latent topics, where each topic is defined by a probability distribution over terms [16,17].

* Corresponding author.
E-mail addresses: carlosgs@unex.es (C. González-Santos), mavega@unex.es (M.A. Vega-Rodríguez), carper@unex.es (C.J. Pérez).

https://doi.org/10.1016/j.knosys.2021.107113
0950-7051/© 2021 Elsevier B.V. All rights reserved.

Some models have been developed by modifying or improving the standard LDA model to attend to more specific tasks [16,18,19].

Models' evaluation is an important task in this context. A model is built up to accomplish an objective. To prove that the new model is better, an evaluation and comparisons with other competitive models must be carried out. There exist some quality measures for model comparison purposes [20,21]. The chosen measure depends on the objective that follows a specific topic model. Thus, some of the main objectives are coverage, coherence, and perplexity. Most of the models are evaluated according to only one objective. This limits their purpose and possible usefulness.

A topic model can provide more accurate and robust results if it considers more than one objective. Then, a multi-objective approach can be considered, which can be solved by applying a multi-objective optimization algorithm. This specific topic has not been properly addressed in the scientific literature. To the best of the authors' knowledge, the only approach addressing this issue from a multi-objective viewpoint was provided by Khalifa et al. [22]. They demonstrated that a multi-objective evolutionary algorithm based on decomposition (MOEA/D [23]) with two objectives performed better than LDA. They also proposed a hybrid approach for three objectives, a more complete approach, that consisted of an initial model given by LDA and applying MOEA/D to the preloaded model, which provided better results.

The goal of this paper is demonstrating the efficiency of a new approach based on a multi-objective artificial bee colony (MOABC) algorithm when addressing the topic modeling from a multi-objective viewpoint. MOABC has been designed and implemented in a so efficient way that it improves the solutions obtained by LDA and MOEA/D approaches. Experiments using several scenarios have been conducted with two well-known datasets in the area of topic modeling (Reuters-21578 [24] and TagMyNews [25]), and the evaluation of the results has been performed with different quality metrics. Therefore, the main contributions of this work are:

- Topic modeling has been addressed from a multi-objective point of view. Although topic modeling needs the optimization of several objectives, this is the second proposal that uses multi-objective optimization, that is, to date, the only multi-objective approach published in the scientific literature is [22], which uses a multi-objective evolutionary algorithm based on decomposition (MOEA/D).
- A new approach based on MOABC has been designed, implemented, and applied to topic modeling. This approach is very different to MOEA/D, because MOABC is based on swarm intelligence and Pareto dominance, and has different operators. The approach has been evaluated by using documents from the Reuters-21578 and TagMyNews datasets.
- In the proposed multi-objective approach, three objective functions have been optimized: coherence, coverage, and perplexity. Besides the values of the three objectives, other three multi-objective quality metrics (hypervolume, set coverage, and distance to the ideal point) have been used for assessment purposes.
- Results show that MOABC provides relevant improvements with respect to LDA (the most used approach for topic modeling) and MOEA/D (the only multi-objective approach published to date).

This paper is organized as follows. In Section 2, the problem is described, including the objectives and the formulation of the multi-objective optimization problem. The methodology of the new approach, including the MOABC algorithm and its operators, is detailed in Section 3. In Section 4, the used datasets, the experimental settings, and the experimental results are presented, with all the comparisons with LDA and MOEA/D. Finally, the conclusions and future work are indicated in Section 5.

2. Problem definition

Let $D = \{D_m\}_{m=1}^{M}$ be a collection of M documents. Each document D_m can be represented as a vector with N_{D_m} terms $w_{D_m} = (w_{D_m 1}, \ldots, w_{D_m N_{D_m}})$. The set of all the terms defines the corpus. Each term of the corpus must be associated to one of the K topics in $T = \{T_k\}_{k=1}^{K}$.

LDA is the most recognized method for topic modeling. LDA assumes that each document can be represented as a probabilistic distribution over latent topics, and that the topic distributions in all documents have a common Dirichlet prior distribution. This model considers a multinomial distribution for D generated from a Dirichlet distribution with parameter α and a multinomial distribution for T generated from a Dirichlet distribution with parameter β. For each term, LDA carries out two steps: firstly, it chooses a topic T_k depending on the multinomial distribution of D; later, a term $w_{D_m n}$ is chosen according to the multinomial distribution of T_k. LDA estimates and maximizes the posterior probability distribution of the observed data D, $p(D|\alpha, \beta)$ [16].

2.1. Objectives

In order to measure the quality of a topic model, it is necessary to evaluate it and make comparisons with others. There are some objectives that can be considered to perform this evaluation. In this paper, the most relevant objectives, that have been used in this knowledge field, are considered, i.e., coverage, coherence, and perplexity. Next, these objectives are described.

Coverage. This objective measures the fraction of corpus covered by the chosen topics. Coverage provides good results if the terms that represent each topic can cover the whole corpus or a relevant part of it. This objective is based on the Euclidean distance between the fraction covered by the topics and the whole corpus.

First of all, it is necessary to calculate the proportion of relevance of each topic T_k for each document D_m. This proportion is calculated as follows:

$$prop(T_k, D_m) = \frac{\sum_{n=1}^{N_{T_k}} tf(w_{T_k n}, D_m)}{N_{T_k} - count_w(T_k, D_m) + 1}, \tag{1}$$

where $tf(w_{T_k n}, D_m)$ is the frequency of the term $w_{T_k n}$ in the document D_m, being $w_{T_k n}$, $n = 1, 2, \ldots, N_{T_k}$, the terms included in the topic T_k. N_{T_k} indicates the number of terms that the topic T_k has and $count_w(T_k, D_m)$ is the number of terms that appear in both topic T_k and document D_m.

The coverage for a single document is obtained by calculating the following formula via Euclidean distance:

$$coverage(D_m) = \sqrt{\sum_{n=1}^{N_{D_m}} \left(\frac{tf(w_{D_m n}, D_m)}{N_{D_m}} - \sum_{k=1}^{K} weight(w_{D_m n}, T_k) prop(T_k, D_m) \right)^2}, \tag{2}$$

where $weight(w_{D_m n}, T_k)$ is the weight that term $w_{D_m n}$ has on the topic T_k. Following the Euclidean distance perspective for each term $w_{D_m n}$, the ideal value with respect to the document D_m is obtained from the term frequency of $w_{D_m n}$ divided by the total number of terms in the document N_{D_m}. The current value with respect to the set of topics is calculated with the summation of every topic T_k, multiplying the term-weight of $w_{D_m n}$ by the proportion of each topic in the document D_m, given by Eq. (1).

Finally, the coverage is defined as:

$$CovObj = \sqrt{\frac{\sum_{m=1}^{M} coverage(D_m)^2}{M}}. \tag{3}$$

A lower $CovObj$ provides a better topic model. Therefore, this objective should be minimized.

Algorithm 1 presents the pseudocode for calculating $CovObj$, giving a more detailed explanation of its computation.

Algorithm 1: Pseudocode for calculating *CovObj*.

```
Input  : $M$: number of documents, $K$: number of topics,
         $N_{T_k}$: number of terms that the topic $T_k$ has, and
         $N_{D_m}$: number of terms that the document $D_m$ has.
Output: $CovObj$: value of the Coverage objective.
$CovObj \leftarrow 0$;
for $m$ = 1 to $M$ do
    for $k$ = 1 to $K$ do
        $total\_tf \leftarrow 0$;
        $count_w \leftarrow 0$;
        for $n$ = 1 to $N_{T_k}$ do
            if $w_{T_k n} \in D_m$ then
                $total\_tf \leftarrow total\_tf + tf(w_{T_k n}, D_m)$;
                $count_w \leftarrow count_w + 1$;
            end
        end
        $prop[k] \leftarrow total\_tf / (N_{T_k} - count_w + 1)$;
    end
    $coverage\_D_m \leftarrow 0$;
    for $n$ = 1 to $N_{D_m}$ do
        $coverage\_topics \leftarrow 0$;
        for $k$ = 1 to $K$ do
            $coverage\_topics \leftarrow$
            $coverage\_topics + (weight(w_{D_m n}, T_k) * prop[k])$;
        end
        $coverage\_ideal \leftarrow tf(w_{D_m n}, D_m)/N_{D_m}$;
        $coverage\_D_m \leftarrow coverage\_D_m +$
        $pow(coverage\_ideal - coverage\_topics, 2)$;
    end
    $coverage\_D_m \leftarrow sqrt(coverage\_D_m)$;
    $CovObj \leftarrow CovObj + pow(coverage\_D_m, 2)$;
end
$CovObj \leftarrow sqrt(CovObj/M)$;
return $CovObj$;
```

Coherence. This objective evaluates the quality of the selected topics in terms of interpretability and semantics [26]. This is done by applying the Pointwise Mutual Information (PMI) method. PMI calculates the coherence of T by scoring, for each topic T_k, its most relevant term pairs, i.e., those terms with the largest weights according to function $weight(w_{D_m n}, T_k)$ from Eq. (2). These scores are obtained by using term co-occurrence. Co-occurrence exists when two terms appear together in a sliding window of a fixed number of terms that goes over the whole corpus. In this way, it can be measured how often terms that co-occur in a topic tend to co-occur in general (in the corpus).

The co-occurrence score for every pair of terms $w_{T_k i}$ and $w_{T_k j}$ is calculated as follows:

$$\text{co-occurrence}(w_{T_k i}, w_{T_k j}) = \begin{cases} -1, & \text{if } P(w_{T_k i}, w_{T_k j}) = 0 \\ \frac{logP(w_{T_k i}) + logP(w_{T_k j})}{logP(w_{T_k i}, w_{T_k j})} - 1, & \text{otherwise,} \end{cases} \tag{4}$$

where $P(w_{T_k i})$ is the probability that $w_{T_k i}$ appears in the corpus, i.e., the number of times this term appears in the corpus divided by the total number of terms in the corpus, and $P(w_{T_k i}, w_{T_k j})$ is the probability that both terms co-occur.

To calculate PMI for a single topic T_k, every pair of relevant terms has to be considered:

$$pmi(T_k) = \frac{\sum_{i=1}^{T_{k_{top}}-1} \sum_{j=i+1}^{T_{k_{top}}} \text{co-occurrence}(w_{T_k i}, w_{T_k j})}{\binom{T_{k_{top}}}{2}}, \tag{5}$$

where $T_{k_{top}}$ is the number of relevant terms that the topic T_k has and $\binom{T_{k_{top}}}{2}$ is the number of possible combinations taken two by two of a set of $T_{k_{top}}$ elements. This combinatorial number is used to avoid the influence of the number of relevant terms selected.

Finally, the coherence is defined as:

$$CoheObj = \sqrt{\sum_{k=1}^{K} (1 - pmi(T_k))^2}. \tag{6}$$

The greater the value of $CoheObj$ is, the worse coherence the topic model has. Then, this objective should be minimized.

Algorithm 2 presents the pseudocode for calculating $CoheObj$, giving a more detailed explanation of its computation.

Algorithm 2: Pseudocode for calculating *CoheObj*.

```
Input  : $K$: number of topics and $T_{k_{top}}$: number of relevant
         terms that the topic $T_k$ has.
Output: $CoheObj$: value of the Coherence objective.
$CoheObj \leftarrow 0$;
for $k$ = 1 to $K$ do
    $pmi\_T_k \leftarrow 0$;
    for $i$ = 1 to $T_{k_{top}} - 1$ do
        for $j$ = i+1 to $T_{k_{top}}$ do
            $Prob_{ij} \leftarrow P(w_{T_k i}, w_{T_k j})$;
            if $Prob_{ij} = 0$ then
                $co\text{-}occurrence_{ij} \leftarrow -1$;
            end
            else
                $co\text{-}occurrence_{ij} \leftarrow$
                $((log(P(w_{T_k i})) + log(P(w_{T_k j})))/log(Prob_{ij})) - 1$;
            end
            $pmi\_T_k \leftarrow pmi\_T_k + co\text{-}occurrence_{ij}$;
        end
    end
    $binomial\_coef \leftarrow (T_{k_{top}} * (T_{k_{top}} - 1))/2$;
    $pmi\_T_k \leftarrow pmi\_T_k / binomial\_coef$;
    $CoheObj \leftarrow CoheObj + pow(1 - pmi\_T_k, 2)$;
end
$CoheObj \leftarrow sqrt(CoheObj)$;
return $CoheObj$;
```

Perplexity. This objective measures the capacity of a previously trained model to discover well-considered topics in unseen documents [27]. To carry out this process, the whole corpus is divided into training and test documents. Training documents are used to train the topic model. Once the model has been trained, it is used to estimate topics in the collection of test documents. Lower results in perplexity are associated to better topic models. Therefore, this objective should be minimized.

The calculation of perplexity is defined in Eq. (7):

$$PerplObj = \frac{-\sum_{m=1}^{M_{test}} logP(w_{D_m} | \mathcal{M})}{10 \sum_{m=1}^{M_{test}} N_{D_m}}, \tag{7}$$

where $\mathcal{M}$ is the topic model used for training and estimating the M_{test} unseen documents. The logarithm of the probabilities is calculated following the "Left-to-Right" algorithm, that uses sequential Monte Carlo methods [27]. Khalifa [28] described an approach based on Left-to-Right for topic modeling.

Algorithm 3 presents the pseudocode for calculating *PerplObj*, giving a more detailed explanation of its computation.

Algorithm 3: Pseudocode for calculating *PerplObj*.

```
Input  : $M_{test}$: number of unseen documents, $\mathcal{M}$: topic model
         to evaluate, and $N_{D_m}$: number of terms that the
         document $D_m$ has.
Output: $PerplObj$: value of the Perplexity objective.
$total\_prob \leftarrow 0$;
$total\_terms \leftarrow 0$;
for $m = 1$ to $M_{test}$ do
    $total\_prob \leftarrow total\_prob + log(P(w_{D_m}|\mathcal{M}))$;
    $total\_terms \leftarrow total\_terms + N_{D_m}$;
end
$PerplObj \leftarrow (-total\_prob)/(10 * total\_terms)$;
return $PerplObj$;
```

2.2. Multi-objective problem formulation

A multi-objective optimization problem maximizes or minimizes a set of objective functions $f_1, f_2, \ldots, f_G$, which defines an objective space. The set of possible solutions x composes the decision space. The optimal solutions are calculated using the dominance criterion [29]. This method takes every solution and compares its objective values with the ones of the rest of solutions. A solution x_i is dominated by other solution x_j if x_j is not worse than x_i in any of the objectives and is better than x_i in, at least, one of the objectives. Those solutions that are non-dominated compose the set of optimal solutions, which can be graphically represented as a Pareto front.

From a multi-objective point of view, the topic modeling problem consists of minimizing the three objectives previously explained, i.e, coverage (f_1), coherence (f_2), and perplexity (f_3). Therefore, it can be formulated as:

$$\underset{x}{minimize}\, f(x) = \{f_1(x), f_2(x), f_3(x)\}. \tag{8}$$

The multi-objective optimization approach will take different LDA models as the input. These LDA models will be improved through the execution of a multi-objective optimization algorithm. The output will be the set of non-dominated solutions generated by this algorithm. Every solution is a topic model, therefore, every solution is composed by all the terms $w_{T_k n}$ included in all the topics T_k of T, together with all their corresponding weights $weight(w_{T_k n}, T_k)$ (the weight that term $w_{T_k n}$ has on that topic T_k).

Every solution is represented by a point in the objective space. The best solution (from the set of non-dominated solutions) is considered to be the one with the minimum Euclidean distance to the ideal point. In this case, the ideal point is $(0, 0, 0)$, since the three objectives should be minimized and cannot be negative.

3. Methodology

In this section, the MOABC algorithm, and its operators, designed and implemented for this specific problem are described. Before that, this section includes the explanation of the preprocessing used and the basic ABC algorithm in which MOABC is based.

3.1. Preprocessing

Topic modeling treats words as entities to make recounts, but these selected words need to have a full meaning. So, it is necessary to work with documents that contain words that have some value and remove the useless ones. Once the document collection is preprocessed, words become terms. The following four preprocessing techniques have been used and sequentially applied to the document collection:

1. Tokenization. First step is to transform every document into plain text. Tokenization takes every word as a token, what makes possible to store them in a data structure for further treatment or analysis. In addition, it allows to avoid several symbols that offer less meaning or nothing at all, such as punctuation signs, asterisks, or parenthesis, among others.
2. Lower cases. Words can be taken as different terms if they are written some times in lower case and other times in upper case. To simplify and treat both words as a unified term, every word in upper case is modified to lower case.
3. Stop-word removal. Some words that appear frequently in the documents with no relevant meaning, such as articles or prepositions, are removed from the document collection.
4. Stemming. This technique reduces inflected or derived words to their root form. So, those words with the same root can be grouped as a single term. The Porter stemming algorithm has been used [30,31].

With the previous preprocessing, each document is transformed into a term collection. Topic modeling establishes a statistical relation between each document and all the topics in order to show how much a topic is (its terms are) present in each document.

3.2. ABC algorithm

ABC (Artificial Bee Colony) algorithm is an optimization algorithm based on the intelligent foraging behavior of honey bee colony, that is, it is based on swarm intelligence [32]. This algorithm has been successfully used in the field of data mining, among other fields [33]. The colony divides its bees by tasks, like exploitation and exploration. There exist three types of bees:

- Employed bees: They keep and improve the best known food sources (solutions).
- Onlooker bees: They observe the dance of employed bees to select which food sources (solutions) are the most efficient ones to be exploited.
- Scout bees: They explore new food sources (solutions). Employed or onlooker bees whose food sources (solutions) are exhausted (cannot be improved more) become scout bees.

Initially, all the bees in the colony (with size *population_size*) are employed bees. There is one solution for each employed bee. Later, the onlooker bees duplicate the size of the colony ($2 \cdot population\_size$), also existing one solution for each onlooker bee. Therefore, the information collected by the employed and onlooker bees is represented by a set of solutions $S = \{S_c\}_{c=1}^{C}$, where C is equal to the sum of the number of employed and onlooker bees, i.e., $2 \cdot population\_size$. Furthermore, every solution has a number of trials to be improved, $S_{c_{trials}}$, that allows the scout bees to determine when a solution S_c is exhausted and needs to be replaced by a new solution. For a more detailed explanation, ABC pseudocode is presented in Algorithm 4.

At the beginning, the solution of every employed bee, which belongs to the first half of the set of solutions S, must be initialized, randomly or from some initial set of solutions (line 1). Once

Algorithm 4: Pseudocode of the ABC algorithm.

```
Input  : population_size: bees in the colony, max_cycles:
         maximum number of cycles, and limit: maximum
         number of trials for a solution.
Output: best_solution: the best solution is returned.
1 init_colony(population_size);
2 for cycles = 1 to max_cycles do
3     send_employed_bees(population_size);
4     calculate_probabilities(population_size);
5     send_onlooker_bees(population_size);
6     send_scout_bees($2 \cdot$ population_size, limit);
7     update_best_solution(best_solution);
8 end
9 return best_solution;
```

the colony has been initialized, the main loop of the algorithm starts its execution until *max_cycles* cycles are reached (lines 2 to 8). For every cycle, each employed bee (line 3) exploits its assigned solution and measures the quality of it. Then, for every employed bee's solution, the probability of being chosen as a good solution is calculated (line 4). Onlooker bees observe the solutions of the employed bees and select the best ones to exploit them (line 5), duplicating the size of the colony. The scout bees, the last type of bees, check every solution S_c (line 6). If the number of trials of a solution ($S_{c_{trials}}$) exceeds or is equal to *limit*, it is replaced by a new solution. The last step of each cycle is to update the best solution, replacing *best_solution* if there is a better one in the current set of solutions S (line 7). Finally, the best solution is returned (line 9) when the maximum number of cycles (*max_cycles*) has been reached.

3.3. MOABC algorithm

In this work, ABC algorithm has been adapted for topic modeling to solve the problem at hand from a multi-objective perspective. This adaptation, which includes problem-aware operators, has been named MOABC (Multi-Objective Artificial Bee Colony) algorithm. MOABC algorithm has been designed and implemented so that every solution represents a set of topics T, that is, all the terms $w_{T_k n}$ included in all the topics T_k of T, together with all their corresponding weights $weight(w_{T_k n}, T_k)$ (the weight that term $w_{T_k n}$ has on that topic T_k). Also, the quality of every solution is evaluated with the three objectives previously explained, justifying the multi-objective nature of the algorithm.

MOABC takes as input the solutions from a number of different LDA models. This number is equal to *population_size*. The output is a set of non-dominated solutions that provide improved results with respect to the original ones. The main steps of MOABC are defined in Algorithm 5.

Firstly, the set of non-dominated solutions sol_{ND} is initialized as an empty set (line 1). Later, each one of the employed bees needs to have a solution assigned, so the first half of the set of solutions S is initialized with different solutions from $models_{LDA}$ (line 2). Then, the main loop of the algorithm is executed until *max_cycles* cycles are reached (lines 3 to 11).

The first step of each cycle is to send the employed bees (line 4). Every employed bee exploits its solution, generating a modified one, by applying the employed bee operator with a probability of modification Pm (see Section 3.4). The modified solution is compared with the original solution by dominance. If the modified solution is better, it replaces the original one, and $S_{c_{trials}}$ is restarted to 0. Otherwise, the number of trials ($S_{c_{trials}}$) is incremented.

Algorithm 5: Pseudocode of the MOABC algorithm.

```
Input  : $models_{LDA}$: several solutions initialized from LDA
         models, population_size: bees in the colony,
         max_cycles: maximum number of cycles, limit:
         maximum number of trials for a solution, and Pm:
         probability of modification.
Output: $sol_{ND}$: set of non-dominated solutions.
1  $sol_{ND} \leftarrow \emptyset$;
2  init_colony(population_size, $models_{LDA}$);
3  for cycles = 1 to max_cycles do
4      send_employed_bees(population_size, Pm);
5      ranking & crowding(population_size);
6      calculate_probabilities(population_size);
7      send_onlooker_bees(population_size);
8      send_scout_bees($2 \cdot$ population_size, limit, $models_{LDA}$);
9      ranking & crowding($2 \cdot$ population_size);
10     update_ND($sol_{ND}$);
11 end
12 return $sol_{ND}$;
```

Once the employed bees have exploited their solutions, these solutions are sorted by ranking and crowding [29] (line 5). First, they are ranked in different sets of solutions (Pareto sets) by dominance. For every solution in the first half of S, the number of solutions dominating that solution is counted. So, those solutions that are not dominated by any solution are ranked in the first set of solutions (first Pareto set), those solutions that are dominated only by solutions in the first Pareto set are ranked in the second set of solutions (second Pareto set), and so on. Later, for every set of solutions (Pareto set), the crowding distance of each solution is calculated. All the solutions in each set are ordered according to the values of their objectives. Then, the distances from each solution to its two nearest solutions, the preceding and the posterior ones, are calculated. Larger crowding distances produce more varied points in the Pareto front, what will lead to a more diverse set of solutions.

Ranking and crowding are performed to classify the solutions by multi-objective quality. Then, every solution is scored by calculating the probability of being chosen as a good solution (line 6). This step simulates the dance of the employed bees to give information about the quality of the food sources they are exploiting.

Later, the onlooker bees observe the solutions from the employed bees and select the best ones to exploit them, generating new solutions in the second half of the solution set S (line 7), and therefore, duplicating the size of the colony ($2 \cdot population\_size$). The new solutions are generated by applying the onlooker bee operator (see Section 3.4). Current and new solution are compared by dominance. If the new solution is not dominated by the current solution, this last one is replaced by the new solution. As it happened with the employed bees, if there is a replacement, the number of trials of the solution ($S_{c_{trials}}$) is restarted to 0. Otherwise, $S_{c_{trials}}$ is incremented.

Scout bees are the last type of bees working in every cycle (line 8). For every solution, a scout bee checks the number of trials. If $S_{c_{trials}}$ reaches the value of *limit*, the current solution is exhausted, so it is replaced by a new one, which is loaded with a solution from $models_{LDA}$, that is randomly selected.

Once all bees have completed their tasks, all the solutions are sorted by quality, that is, by ranking and crowding again (line 9). The last step of the cycle is to update sol_{ND} with the non-dominated solutions (line 10). Furthermore, the first half (the best solutions) of the current set of solutions S (a total of *population_size* solutions) will be the employed bees for the next cycle, and the process is repeated again.

When *cycles* reaches its maximum value (*max_cycles*), the set of non-dominated and improved solutions, sol_{ND}, is returned (line 12).

3.4. MOABC operators

Two operators have been mentioned in the MOABC algorithm: employed bee operator and onlooker bee operator. The employed bee operator seeks to improve its solution by modifying the weight of a part of the term collection from each topic. Under a probability of modification *Pm*, the operator tries to modify every term of every topic. For every term $w_{T_k n}$, the operator selects a random number in the range [0,1). If it is less than *Pm*, the weight of the term is modified. More specifically, it is incremented in an amount proportional to its own weight. This amount is calculated by selecting a random percentage (0,100] of the term weight and adding it to itself. Fig. 1 shows in detail how this operator works over a single topic in a basic example. The green arrow indicates that the randomly generated number from [0,1) for the term is less than *Pm*. The red arrow indicates the opposite, so the term weight is not modified. For example, the weight of $w_{T_k 3}$ is 0.015 and it is incremented by the 46.67% of its weight, so its final weight is 0.022.

For the onlooker bee operator, every onlooker bee modifies its solution taking into account another solution that is randomly selected from the first set of solutions (first Pareto set) after the ranking and crowding operations. Firstly, one of the two solutions (the original one or the selected one) is randomly chosen to establish the number of terms N_{T_k} that the new solution is going to take for every topic. Then, for every topic, the established number of terms are randomly selected, one by one, from one of the two solutions. For every term, if both solutions have that term in the same topic, the term weight is the mean of the weights in both solutions. Otherwise, it just takes the weight of the chosen term. Fig. 2 explains how this operator works over a single topic. In this basic example, the first topic (T_1) of solutions S_1 and S_2 is used for generating the first topic of the new solution S_{new}. The terms "develop" and "credit" are in S_1, but they are not in S_2; and the terms "govern" and "transport" are in S_2, but they are not in S_1. So these terms keep their weights if they are randomly chosen by S_{new}. The terms "plastic" and "bank" are present in both solutions, so the weight mean is calculated for these terms.

4. Results and comparisons

In this section, the datasets used for the experiments are presented. Then, the experimental settings are described. Finally, the results and comparisons for the three objective functions and other three multi-objective quality metrics are shown, in addition to the corresponding Pareto fronts. The MOABC results will be compared with the results from LDA (the most used approach for topic modeling) and MOEA/D [22] (the only multi-objective approach for topic modeling published to date).

4.1. Datasets

The experiments have been performed by using the datasets: Reuters-21578 [24] and TagMyNews [25]. These datasets have been previously used in text analysis, for example, for text classification [34], feature selection [35], influencer detection [36], document clustering [37], and topic modeling [38]. They contain documents of short to medium length. On the one hand, for the Reuters-21578 dataset, the document collection is classified with different tags, such as topics, organizations, places, people, and exchanges. For this problem, the documents need to be tagged with at least one topic. Therefore, the documents have been filtered by considering documents meeting this requirement. After filtering, the final collection has 10,211 documents. On the other hand, for the TagMyNews dataset, the document collection consists of documents containing news titles and a short description of the news. These documents are categorized in 7 different categories: world, US, science and technology, sport, business, health, and entertainment. In this dataset, all the documents have passed the filtering. Therefore, after filtering, the final collection has 32,604 documents (the same amount as the original collection). Furthermore, the datasets have been preprocessed, by following the steps in Section 3.1. Table 1 shows some information about the datasets.

Table 1
Main characteristics of the Reuters-21578 and TagMyNews datasets.

Characteristics	Counts
Reuters-21578 dataset	
Number of documents	21,578
Number of documents after filtering	10,211
Number of different words after filtering	31,525
Average document length (in words) after filtering	≃ 130
Number of different terms after filtering and preprocessing	19,238
Average preprocessed document length (in terms) after filtering	≃ 70
TagMyNews dataset	
Number of documents	32,604
Number of documents after filtering	32,604
Number of different words after filtering	35,469
Average document length (in words) after filtering	≃ 33
Number of different terms after filtering and preprocessing	23,310
Average preprocessed document length (in terms) after filtering	≃ 19

4.2. Experimental settings

In the experiments, all the approaches (LDA, MOEA/D, and MOABC) use the same configuration for the number of topics and other basic parameters. Two scenarios (4 and 10 topics) have been considered to compare with LDA and MOEA/D. These two scenarios are the same that were considered by [22] with MOEA/D. The sliding window and $T_{k_{top}}$ have been set to 10 for the computation of the coherence objective (see Section 2.1). The perplexity works with two sets of documents for training and testing. The training set corresponds to the 90% of the document collection, whereas the remaining 10% is assigned to the testing set.

LDA needs to set two parameters before its execution: α and β, where $\alpha = 50/K$, being K the number of topics, and $\beta = 0.01$ [22,28]. By the other hand, MOEA/D [22] needs to set the population size and the maximum number of cycles, which are 45 and 50, respectively.

Besides, in order to make a fair comparison, MOABC uses the same population size and maximum number of cycles as MOEA/D. Nevertheless, MOABC has two additional parameters (*limit* or number of trials and probability of modification *Pm*), which have been adjusted after a parametric study to $limit = 4$ and $Pm = 0.07$. As can be observed in Table 2, different values have been tested for these two parameters, highlighting in gray the best configuration found.

The results shown in the following subsections are the outcome from 31 independent runs performed for each experiment in order to ensure reliable statistics. These experiments have been performed on a computer with an Intel Core i5-5200U CPU with 8GB RAM, and operating system Windows 10. MOABC and MOEA/D have been implemented in C/C++ with the CodeBlocks 17.12 IDE. For LDA, the toolkit MALLET 2.0.8 [39] has been used.

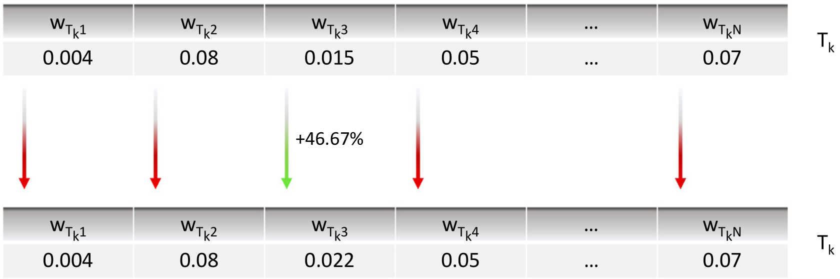


Fig. 1. Illustration of how the employed bee operator works over a single topic in a simple example.

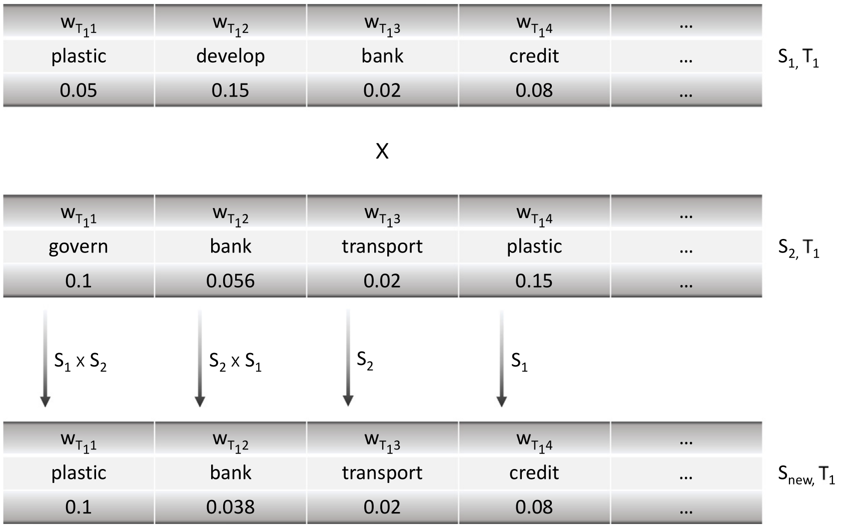


Fig. 2. Illustration of how the onlooker bee operator works over a single topic in a simple example.

Table 2
All tested values to find the best MOABC configuration. The best value for each parameter appears gray shadowed.

	Checked values								
Pm	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09
limit	2	3	4	5	6	7	10	15	

4.3. Results for the objective functions

Each one of the three objective functions (coherence, coverage, and perplexity) has been calculated as presented in Section 2.1. Table 3 shows the mean and standard deviation of the best values for the three objectives obtained by the three approaches (MOABC, LDA, and MOEA/D), under the two considered scenarios with 4 and 10 topics.

MOABC presents important improvements for coherence in comparison with LDA and MOEA/D. The coverage objective shows a more similar situation between both multi-objective approaches (MOABC and MOEA/D), whereas LDA shows the worst results. For the last objective, perplexity, MOABC is the best approach again.

As a summary, taking into account the three objectives, MOABC shows a clear improvement with respect to LDA, the most used approach in the field of topic modeling. In comparison with MOEA/D [22], the only multi-objective proposal to date for topic modeling, MOABC outperforms it in 10 out of 12 cases.

In the next subsections, quality of the solutions and other aspects related to multi-objective optimization will be presented and discussed.

Table 3
Mean and standard deviation ($mean_{\pm standard\_deviation}$) of the objective values obtained from the 31 repetitions for MOABC, LDA, and MOEA/D. The best values appear gray shadowed.

Objective	# topics	MOABC	LDA [39]	MOEA/D [22]
Reuters-21578 dataset				
Coherence	4	0.98580$_{\pm0.027}$	1.03308$_{\pm0.032}$	1.13560$_{\pm0.102}$
	10	1.69918$_{\pm0.013}$	1.76422$_{\pm0.036}$	1.73156$_{\pm0.033}$
Coverage	4	0.27835$_{\pm0.000}$	0.27914$_{\pm0.000}$	0.27768$_{\pm0.001}$
	10	0.27452$_{\pm0.000}$	0.27689$_{\pm0.000}$	0.27493$_{\pm0.000}$
Perplexity	4	0.77387$_{\pm0.002}$	0.81590$_{\pm0.016}$	0.78270$_{\pm0.007}$
	10	0.77667$_{\pm0.003}$	0.83251$_{\pm0.006}$	0.77678$_{\pm0.003}$
TagMyNews dataset				
Coherence	4	1.74097$_{\pm0.187}$	1.89456$_{\pm0.051}$	1.74349$_{\pm0.349}$
	10	2.16842$_{\pm0.432}$	2.23615$_{\pm0.112}$	2.49123$_{\pm0.427}$
Coverage	4	0.27015$_{\pm0.000}$	0.27017$_{\pm0.000}$	0.27015$_{\pm0.000}$
	10	0.27009$_{\pm0.000}$	0.27011$_{\pm0.000}$	0.27010$_{\pm0.000}$
Perplexity	4	0.86217$_{\pm0.009}$	0.89394$_{\pm0.001}$	0.87988$_{\pm0.015}$
	10	0.88870$_{\pm0.010}$	0.89228$_{\pm0.001}$	0.89055$_{\pm0.005}$

4.4. Hypervolume indicator

The hypervolume (HV) indicator is one of the most well-known metrics to evaluate multi-objective approaches [40]. The hypervolume indicator measures the k-dimensional space covered among the solution points of a Pareto front and the worst possible point (taken as a reference). Given a point set $\mathcal{A} \subset \mathbb{R}^k$

Table 4
Median and quartile deviation ($median_{\pm quartile\_deviation}$) obtained from the 31 hypervolume values for MOABC, LDA, and MOEA/D. The best values appear gray shadowed.

# topics	MOABC	LDA [39]	MOEA/D [22]
Reuters-21578 dataset			
4	31.21%±0.03%	0.74%±0.01%	28.60±0.08%
10	63.48%±0.08%	2.77%±0.01%	44.37±0.01%
TagMyNews dataset			
4	28.22%±1.15%	3.12%±1.86%	20.27±1.14%
10	17.56%±0.79%	4.42%±1.63%	13.55±0.66%

Table 5
Set coverage based on the median Pareto fronts for MOABC, LDA [39], and MOEA/D [22]. The best values appear gray shadowed.

Reuters-21578 dataset		
# topics	SC(MOABC, LDA)	SC(LDA, MOABC)
4	100.00%	0.00%
10	100.00%	0.00%
# topics	SC(MOABC, MOEA/D)	SC(MOEA/D, MOABC)
4	47.89%	18.75%
10	92.16%	0.00%
TagMyNews dataset		
# topics	SC(MOABC, LDA)	SC(LDA, MOABC)
4	100.00%	0.00%
10	100.00%	0.00%
# topics	SC(MOABC, MOEA/D)	SC(MOEA/D, MOABC)
4	52.81%	4.88%
10	88.76%	9.52%

and a reference point $r \in \mathbb{R}^k$, the hypervolume of the point set $\mathcal{A}$ with respect to r is defined as:

$$HV(\mathcal{A}, r) = \mathcal{L}\left(\bigcup_{a \in \mathcal{A}} \{b | a \succeq b \succeq r\}\right), \tag{9}$$

where $\mathcal{L}(\cdot)$ denotes the Lebesgue measure [41] and $a \succeq b$ means that a dominates b. A more detailed explanation of this equation can be found in [40].

Since these approaches have three objectives, the hypervolume indicator works in a three-dimensional space. The values of the objective functions have been normalized in the range [0, 1], so r is (1, 1, 1). Table 4 shows the median and quartile deviation of the hypervolume obtained by the three approaches (MOABC, LDA, and MOEA/D), under the two considered scenarios with 4 and 10 topics. Higher values are better. It can be appreciated how the multi-objective approaches take a relevant difference with respect to LDA for both scenarios. Besides, MOABC clearly outperforms MOEA/D [22] in both scenarios.

4.5. Pareto front analysis

Fig. 3 (for the Reuters-21578 dataset) and Fig. 4 (for the TagMyNews dataset) show the two-dimensional projections of the median Pareto fronts for MOABC, MOEA/D, and LDA. They correspond to the Pareto fronts associated with the median hypervolume of the 31 repetitions.

Recalling that all objective functions have to be minimized, it can be observed how MOABC presents the lowest values, supporting the improvement of this new approach with respect to MOEA/D [22] and LDA. Furthermore, it can be appreciated that the objectives are conflicting each other. For example, in Figs. 3(c,d) and 4(c,d), it can be seen that, in general, better (lower) values of coverage correspond to worse (higher) values of perplexity, and vice versa. In the same way, in Fig. 3(e,f) and 4(e,f), it can be noted that, in general, better (lower) values of perplexity correspond to worse (higher) values of coherence, and vice versa. Regarding LDA, there is only one point per graphic because only one solution is obtained per execution. LDA points are very distant from the ones of the other two approaches (and further away from the best theoretical point), justifying how multi-objective optimization works better.

4.6. Set coverage indicator

The set coverage (SC) indicator is a metric that performs a comparison between two Pareto fronts $\mathcal{A}$ and $\mathcal{B}$. $SC(\mathcal{A}, \mathcal{B})$ measures the percentage of solutions from $\mathcal{B}$ that are covered ($\succeq$) by solutions from $\mathcal{A}$. Note that SC is not a symmetrical indicator, so it is necessary to calculate $SC(\mathcal{A}, \mathcal{B})$ and $SC(\mathcal{B}, \mathcal{A})$. The following equation shows how the set coverage is calculated:

$$SC(\mathcal{A}, \mathcal{B}) = \frac{|\{b_j \in \mathcal{B}; \exists a_i \in \mathcal{A} : a_i \succeq b_j\}|}{|\mathcal{B}|}, \tag{10}$$

where $|\mathcal{B}|$ is the cardinality of the set $\mathcal{B}$.

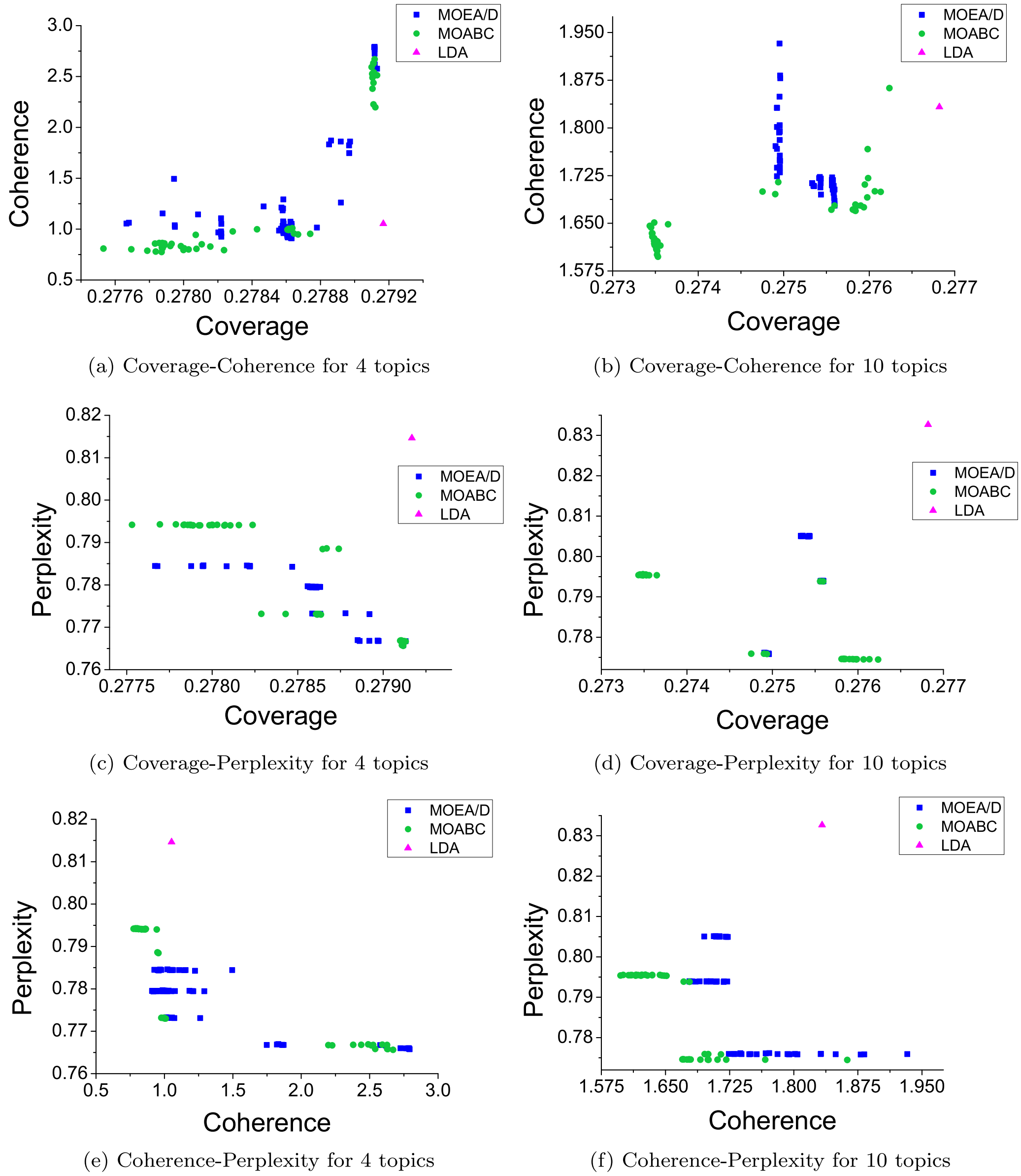
The set coverage has been executed with the median Pareto fronts obtained for MOABC, LDA, and MOEA/D. Table 5 presents, in percentages, the set coverage calculated for MOABC and one of the other two approaches each time.

MOABC outperforms LDA in both scenarios, since the LDA solutions are covered by the MOABC solutions and none of the MOABC solutions is covered by the LDA solutions. MOABC is also clearly better than MOEA/D [22]. For the results of MOABC and MOEA/D in the 4 topics' scenario, around the half of the MOEA/D solutions (47.89% for Reuters-21578 and 52.81% for TagMyNews) are covered by the MOABC solutions, whereas only a few MOABC solutions (18.75% for Reuters-21578 and 4.88% for TagMyNews) are covered by the MOEA/D ones. Furthermore, MOABC greatly is the best with 10 topics, where most of the MOEA/D solutions (92.16% for Reuters-21578 and 88.76% for TagMyNews) are covered by the MOABC solutions and only a few MOABC solutions (none for Reuters-21578 and 9.52% for TagMyNews) are covered by the MOEA/D solutions.

4.7. Distance to the ideal point

The last considered multi-objective metric is the distance to the ideal point (DIP). This metric takes all the solutions from a Pareto front and measures the Euclidean distance between every solution and the ideal point. The nearest point to the ideal one is taken as the best solution.

For every approach, DIP has been calculated for the 31 Pareto fronts. The objective values have been normalized in the range [0, 1] to balance the impact of each objective. As all the objective functions have to be minimized, the ideal point has been set to (0, 0, 0). Table 6 shows the median and quartile deviation for the 31 DIPs for every approach (MOABC, LDA, and MOEA/D) in the two scenarios. It can be observed that LDA presents the worst results again, very distant from the multi-objective approaches. MOABC offers the best results for both scenarios with an important difference with respect to MOEA/D [22]. Briefly, MOABC clearly has the least distant solutions with respect to the ideal solution in all the cases.



(a) Coverage-Coherence for 4 topics

(b) Coverage-Coherence for 10 topics

(c) Coverage-Perplexity for 4 topics

(d) Coverage-Perplexity for 10 topics

(e) Coherence-Perplexity for 4 topics

(f) Coherence-Perplexity for 10 topics

Fig. 3. Two-dimensional projections of the median Pareto fronts for MOABC, MOEA/D [22], and LDA [39]. Experiments have been performed with 4 (a,c,e) and 10 topics (b,d,f) using the Reuters-21578 dataset.

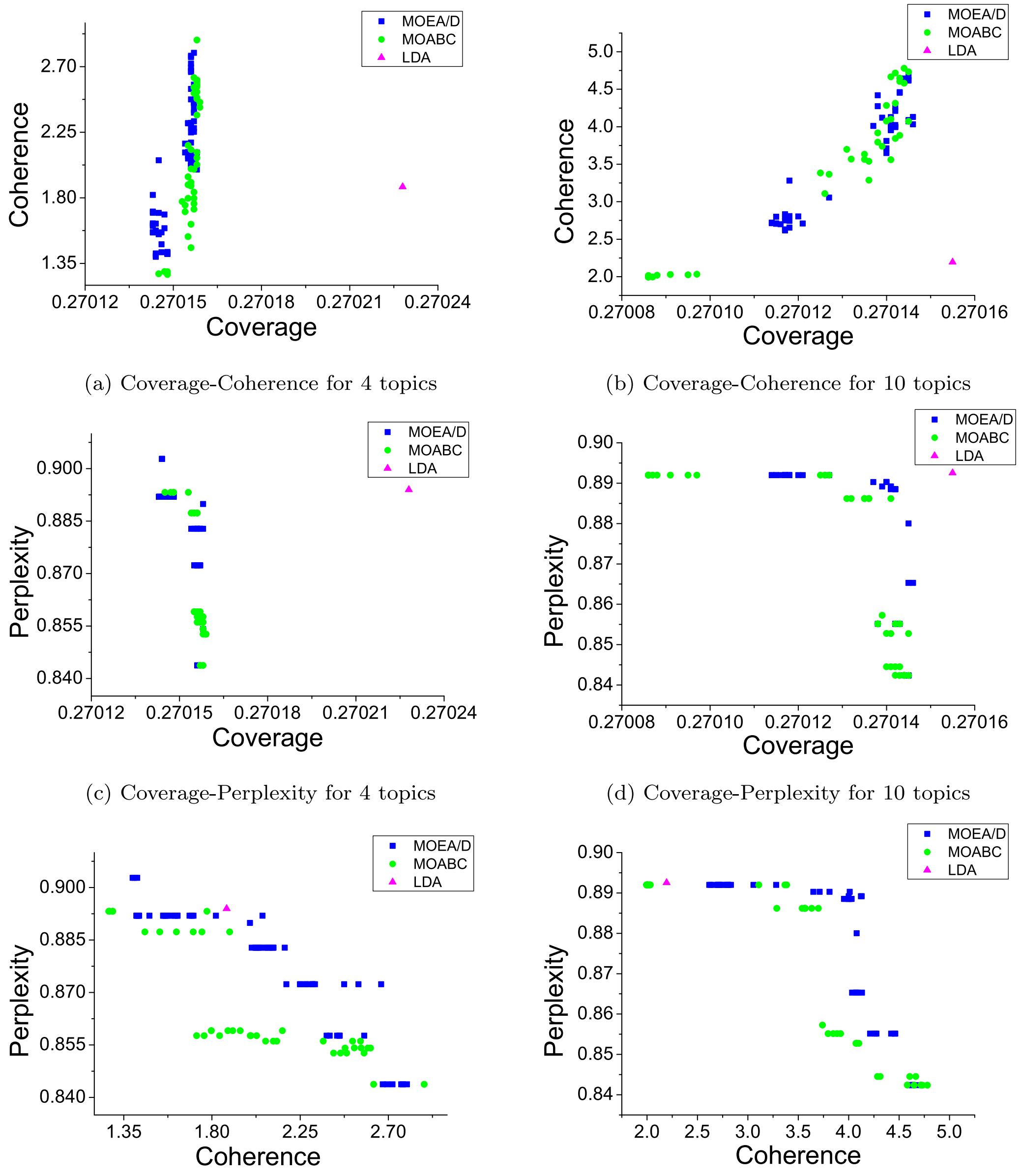
Table 6
Median and quartile deviation ($median_{\pm quartile\_deviation}$) of the 31 *DIP* values (repetitions) for MOABC, LDA, and MOEA/D. The best values appear gray shadowed.

# topics	MOABC	LDA [39]	MOEA/D [22]
Reuters-21578 dataset			
4	0.65497±0.062	1.17467±0.094	0.71539±0.119
10	0.38890±0.047	1.17088±0.033	0.53488±0.008
TagMyNews dataset			
4	0.70571±0.021	1.14293±0.111	0.83622±0.024
10	0.88991±0.028	1.08189±0.077	1.00488±0.029

4.8. What is the price of multi-objective optimization algorithms?

MOABC (the proposed approach) and MOEA/D [22] obtain better performance than LDA [39] in all the comparisons, showing the advantage of multi-objective optimization algorithms in topic modeling. However, the question is: what is the price of multi-objective optimization algorithms? This question can be answered from two points of view.

On the one hand, from the viewpoint of space complexity, MOABC and MOEA/D use some additional memory space because they have to store the information of a population of solutions. Fig. 1 shows the main information stored in every solution. As it can be seen, every solution includes an index to every term included in every topic and the weight of every term. In our implementation in C/C++, every term index is implemented with the data type “int” (4 bytes) and every term weight is implemented with the data type “double” (8 bytes), that is, every term implies 12 bytes. As shown in Table 1, for the Reuters-21578 dataset, the number of terms after filtering and preprocessing is 19,238. Therefore, supposing that all terms are used, a solution implies $19,238 \cdot 12$ bytes = 230,856 bytes. MOABC and MOEA/D have a population size of 45 solutions (see Section 4.2), therefore, the extra memory space for the whole solution population is $45 \cdot 230,856$ bytes = 10,388,520 bytes, that is, around 10 megabytes. As MOABC and MOEA/D have the same space complexity, the



(a) Coverage-Coherence for 4 topics

(b) Coverage-Coherence for 10 topics

(c) Coverage-Perplexity for 4 topics

(d) Coverage-Perplexity for 10 topics

(e) Coherence-Perplexity for 4 topics

(f) Coherence-Perplexity for 10 topics

Fig. 4. Two-dimensional projections of the median Pareto fronts for MOABC, MOEA/D [22], and LDA [39]. Experiments have been performed with 4 (a,c,e) and 10 topics (b,d,f) using the TagMyNews dataset.

conclusion is that it is better to use MOABC because it obtains better performance than MOEA/D in all the comparisons. In the case of LDA, it uses around 10 megabytes less, but its results are clearly worse in all the comparisons, therefore, the conclusion is that it is also better to use MOABC because 10 megabytes is not a significant amount of memory. In the case of the TagMyNews dataset, the analysis is very similar, with the only difference that the number of terms after filtering and preprocessing is 23,310 (see Table 1). Therefore, in this dataset, following the same calculations, the additional memory space is 12,587,400 bytes, that is, around 13 megabytes, which is again a non-significant amount of memory.

On the other hand, from the viewpoint of time complexity, in average for both datasets, the execution time of MOABC and MOEA/D is around 1 h (e.g. MOABC has an average runtime of 46 min for 4 topics and 59 min for 10 topics, in the TagMyNews dataset, which is the dataset with more documents and words/terms), while the execution time of LDA is around 1 min. These execution times depend on the hardware and software used. In this work, the hardware and software used are the ones indicated at the end of Section 4.2, which are not the fastest ones (CPU i5-5200U and Windows 10). As MOABC and MOEA/D have the same time complexity, the conclusion is that it is better to use MOABC because it obtains better performance than MOEA/D in all the comparisons. In the case of LDA, its execution time is clearly lower, but its results are clearly worse in all the comparisons, therefore, the decision is not easy. It could depend on the most important parameter for the user in each case: result quality or time. One advantage of multi-objective optimization algorithms is that they are highly parallelizable. For example, MOABC uses a population of solutions and the computations performed over every solution are independent on the computations performed over the other solutions, that is, there are not data dependencies. In this way, the loops "for" performing the computations over every solution can be parallelized. In this work, MOABC has a population size of 45 solutions, therefore, the peak speed-up would be 45, that is, after parallelization, MOABC could be up to 45 times faster, reducing its execution time to around 1 or 2 min. For the parallelization of these loops "for", OpenMP [42] is a very good alternative. OpenMP is one of the most used standards for parallel programming in the widely-present multi-core architectures. Observe that the possible parallelization of LDA does not produce the same benefits because its current execution time is already small. In conclusion, comparing a possible parallel MOABC with LDA, the decision would be easier, because their execution times will be more similar, and the LDA's results are clearly worse in all the comparisons, therefore, the conclusion would be that it is better to use MOABC. Hence, the parallelization of MOABC is a very interesting line of future work.

5. Conclusions and future work

Topic modeling problem has been traditionally solved with single-objective approaches, despite the fact that it is a problem that can be better addressed with multi-objective approaches. In this work, a multi-objective optimization approach based on the swarm intelligence of a bee colony (MOABC) has been designed, implemented, and applied to improve the LDA model, the most used approach for topic modeling. The new approach has been explained in detail in this paper, also including the description of its new problem-aware operators. In the multi-objective context, only one approach, MOEA/D [22], has been previously applied to topic modeling. MOABC, LDA, and MOEA/D have been compared according to the three objective functions (coherence, coverage, and perplexity) over different scenarios using Reuters-21578 and TagMyNews datasets, two popular document collections categorized by topics. MOABC surpasses LDA in all the cases due to the fact that MOABC uses three relevant criteria that simultaneously address the optimization of the quality of the selected topics in terms of interpretability and semantics, the measurement of the corpus covered by the chosen topics, and the capacity of the model to discover well-considered topics in unseen documents. Simultaneously using these three criteria provides more information about the topic distribution than by representing the documents as mixtures of latent topics. By the other hand, MOABC outperforms MOEA/D in 10 out of 12 cases. Since, in these approaches, the same three criteria have been used, the superiority of MOABC is due to the way that the multi-objective optimization approach has been defined (algorithmic design and swarm intelligence), and specially to the efficiency of the new problem-aware operators.

In order to perform a thorough comparison, the different Pareto fronts have been analyzed and other three well-known multi-objective quality metrics (hypervolume, set coverage, and distance to the ideal point) have been included in the comparison in the different scenarios. In all the cases, results show that MOABC outperforms both LDA and MOEA/D, and justify the multi-objective nature of the problem.

As future research, MOABC will be parallelized, improving its execution time. More specifically, OpenMP will be used to parallelize the loops "for" that perform the computations over the different solutions in the population. This will speed up the MOABC execution in the current multi-core architectures. On the other side, it could be explored the use of other criteria in MOABC. A topic of interest, that should be addressed, is the definition and adaptation (e.g. based on TF-IDF) of other criteria that could be combined with coherence, coverage, and/or perplexity to, eventually, produce better results. Also, since MOABC has obtained good results solving the topic modeling problem, it could be used to solve other multi-objective problems in the field of natural language processing. Furthermore, from an application point of view, this approach (MOABC) will be adapted to work in a media environment, specifically, to discover topics from document collections of film metadata.

CRediT authorship contribution statement

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

Declaration of competing interest

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

Acknowledgments

This research has been supported by Ministry of Science, Innovation and Universities – Spain and State Research Agency – Spain (Projects PID2019-107299GB-I00/AEI/10.13039/501100011033 and MTM2017-86875-C3-2-R), Junta de Extremadura – Spain (Projects GR18090 and GR18108), and European Union (European Regional Development Fund).

References

[1] C. Wang, D.M. Blei, Collaborative topic modeling for recommending scientific articles, in: Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Association for Computing Machinery, New York, NY, USA, 2011, pp. 448–456, http://dx.doi.org/10.1145/2020408.2020480.

[2] K.R. Chowdhary, Natural language processing, in: Fundamentals of Artificial Intelligence, Springer, New Delhi, India, 2020, pp. 603–649, http://dx.doi.org/10.1007/978-81-322-3972-7_19.

[3] 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) 1–34, http://dx.doi.org/10.3390/bdcc4010001.

[4] J. Liu, C. Liu, N.J. Belkin, Personalization in text information retrieval: A survey, J. Assoc. Inf. Sci. Technol. 71 (3) (2020) 349–369, http://dx.doi.org/10.1002/asi.24234.

[5] G.R. Kumar, S.R. Basha, S.B. Rao, A summarization on text mining techniques for information extracting from applications and issues, J. Mech. Continua Math. Sci. Special Issue 5 (2020) 324–332, http://dx.doi.org/10.26782/jmcms.spl.5/2020.01.00026.

[6] 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.

[7] I. Vayansky, S.A. Kumar, A review of topic modeling methods, Inf. Syst. 94 (2020) 101582, http://dx.doi.org/10.1016/j.is.2020.101582.

[8] S. Mifrah, E.H. Benlahmar, Topic modeling coherence: A comparative study between LDA and NMF models using COVID'19 corpus, Int. J. Adv. Trends Comput. Sci. Eng. 9 (4) (2020) 5756–5761, http://dx.doi.org/10.30534/ijatcse/2020/231942020.

[9] G. L'Huillier, A. Hevia, R. Weber, S. Ríos, Latent semantic analysis and keyword extraction for phishing classification, in: 2010 IEEE International Conference on Intelligence and Security Informatics, 2010, pp. 129–131, doi:10.1109/ISI.2010.5484762.

[10] A. Ianina, K. Vorontsov, Regularized multimodal hierarchical topic model for document-by-document exploratory search, in: 2019 25th Conference of Open Innovations Association (FRUCT), 2019, pp. 131–138, doi:10.23919/FRUCT48121.2019.8981493.

[11] W. Wang, B. Guo, Y. Shen, H. Yang, Y. Chen, X. Suo, Twin labeled LDA: A supervised topic model for document classification, Appl. Intell. 50 (12) (2020) 4602–4615, http://dx.doi.org/10.1007/s10489-020-01798-x.

[12] B. Ozyurt, M.A. Akcayol, A new topic modeling based approach for aspect extraction in aspect based sentiment analysis: SS-LDA, Expert Syst. Appl. 168 (2021) 114231, http://dx.doi.org/10.1016/j.eswa.2020.114231.

[13] E.D. Gutiérrez, E. Shutova, P. Lichtenstein, G. de Melo, L. Gilardi, Detecting cross-cultural differences using a multilingual topic model, Trans. Assoc. Comput. Linguist. 4 (2016) 47–60, http://dx.doi.org/10.1162/tacl_a_00082.

[14] D.M. Blei, J.D. McAuliffe, Supervised topic models, in: Proceedings of the 20th International Conference on Neural Information Processing Systems, in: NIPS'07, Curran Associates Inc., Red Hook, NY, USA, 2007, pp. 121–128.

[15] N.E. Evangelopoulos, Latent semantic analysis, Wiley Interdiscip. Rev. Cogn. Sci. 4 (6) (2013) 683–692, http://dx.doi.org/10.1002/wcs.1254.

[16] H. Jelodar, Y. Wang, C. Yuan, X. Feng, X. Jiang, Y. Li, L. Zhao, Latent Dirichlet allocation (LDA) and topic modeling: Models, applications, a survey, Multimedia Tools Appl. 78 (11) (2019) 15169–15211, http://dx.doi.org/10.1007/s11042-018-6894-4.

[17] M. Anandarajan, C. Hill, T. Nolan, Probabilistic topic models, in: Practical Text Analytics: Maximizing the Value of Text Data, Springer International Publishing, Cham, Switzerland, 2019, pp. 117–130, http://dx.doi.org/10.1007/978-3-319-95663-3_8.

[18] W. Sriurai, Improving text categorization by using a topic model, Adv. Comput. 2 (6) (2011) 21–27, http://dx.doi.org/10.5121/acij.2011.2603.

[19] R. Krestel, P. Fankhauser, W. Nejdl, Latent Dirichlet allocation for tag recommendation, in: Proceedings of the Third ACM Conference on Recommender Systems, in: RecSys '09, Association for Computing Machinery, New York, NY, USA, 2009, pp. 61–68, http://dx.doi.org/10.1145/1639714.1639726.

[20] D. Newman, S. Karimi, L. Cavedon, External evaluation of topic models, in: Australasian Document Computing Symposium, 2009, pp. 11–18.

[21] J. Chang, J. Boyd-Graber, S. Gerrish, C. Wang, D.M. Blei, Reading tea leaves: How humans interpret topic models, in: Proceedings of the 22nd International Conference on Neural Information Processing Systems, in: NIPS'09, Curran Associates Inc., Red Hook, NY, USA, 2009, pp. 288–296.

[22] O. Khalifa, D.W. Corne, M. Chantler, F. Halley, Multi-objective topic modeling, in: 7th International Conference on Evolutionary Multicriterion Optimization, Sheffield, UK, 2013, pp. 51–65, http://dx.doi.org/10.1007/978-3-642-37140-0_8.

[23] Q. Zhang, H. Li, MOEA/D: a multiobjective evolutionary algorithm based on decomposition, IEEE Trans. Evol. Comput. 11 (6) (2007) 712–731, http://dx.doi.org/10.1109/TEVC.2007.892759.

[24] D.D. Lewis, Reuters-21578, Distribution 1.0, 1997, http://archive.ics.uci.edu/ml, (visited on 2 April 2021).

[25] V. Nandwani, Tag my news, 2017, https://github.com/vijaynandwani/News-Classification/blob/master/news, (visited on 2 April 2021).

[26] D. Newman, J.H. Lau, K. Grieser, T. Baldwin, Automatic evaluation of topic coherence, in: Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Association for Computational Linguistics, Los Angeles, California, 2010, pp. 100–108.

[27] H.M. Wallach, I. Murray, R. Salakhutdinov, D. Mimno, Evaluation methods for topic models, in: Proceedings of the 26th Annual International Conference on Machine Learning, Association for Computing Machinery, New York, NY, USA, 2009, pp. 1105–1112, http://dx.doi.org/10.1145/1553374.1553515.

[28] O. Khalifa, New Sampling and Optimization Methods for Topic Inference and Text Classification, (Ph.D. thesis), School of Mathematical and Computer Sciences, Heriot-Watt University, Edinburgh, Scotland, UK, 2018.

[29] 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.

[30] M.F. Porter, The porter stemming algorithm, 2020, http://www.tartarus.org/martin/PorterStemmer/, (visited on 2 April 2021).

[31] M.F. Porter, An algorithm for suffix stripping, Program Electron. Libr. Inf. Syst. 14 (3) (1980) 130–137, http://dx.doi.org/10.1108/eb046814.

[32] D. Karaboga, B. Basturk, A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm, J. Global Optim. 39 (3) (2007) 459–471, http://dx.doi.org/10.1007/s10898-007-9149-x.

[33] D. Karaboga, B. Gorkemli, C. Ozturk, N. Karaboga, A comprehensive survey: artificial bee colony (ABC) algorithm and applications, Artif. Intell. Rev. 42 (1) (2014) 21–57, http://dx.doi.org/10.1007/s10462-012-9328-0.

[34] T. Dogan, A.K. Uysal, A novel term weighting scheme for text classification: TF-MONO, J. Informetr. 14 (4) (2020) 101076, http://dx.doi.org/10.1016/j.joi.2020.101076.

[35] V.N. Thatha, A.S. Babu, D. Haritha, An enhanced feature selection for text documents, in: S.C. Satapathy, V. Bhateja, J.R. Mohanty, S.K. Udgata (Eds.), Smart Intelligent Computing and Applications, Springer Singapore, Singapore, 2020, pp. 21–29, http://dx.doi.org/10.1007/978-981-32-9690-9_3.

[36] T.-T. Quan, D.-T. Mai, T.-D. Tran, CID: Categorical influencer detection on microtext-based social media, Online Inf. Rev. 44 (5) (2020) 1027–1055, http://dx.doi.org/10.1108/OIR-02-2019-0062.

[37] R. Lakshmi, S. Baskar, Efficient text document clustering with new similarity measures, Int. J. Bus. Intell. Data Min. 18 (1) (2021) 49–72, http://dx.doi.org/10.1504/IJBIDM.2021.111741.

[38] P. Xie, E.P. Xing, Integrating document clustering and topic modeling, in: Proceedings of the Twenty-Ninth Conference on Uncertainty in Artificial Intelligence (UAI'13), Arlington, Virginia, USA, 2013, pp. 694–703, arXiv:1309.6874.

[39] A.K. McCallum, MALLET: A MAchine learning for language toolkit, 2018, http://mallet.cs.umass.edu, (visited on 2 April 2021).

[40] K. Shang, H. Ishibuchi, L. He, L.M. Pang, A survey on the hypervolume indicator in evolutionary multi-objective optimization, IEEE Trans. Evol. Comput. 25 (1) (2021) 1–20, http://dx.doi.org/10.1109/TEVC.2020.3013290.

[41] S. Gentili, The lebesgue measure, in: Measure, Integration and a Primer on Probability Theory: Volume 1, Springer International Publishing, Cham, Switzerland, 2020, pp. 197–240, http://dx.doi.org/10.1007/978-3-030-54940-4_9.

[42] T.G. Mattson, Y.H. He, A.E. Koniges, The OpenMP Common Core: Making OpenMP Simple Again, MIT Press, 2019.