I.J. Information Technology and Computer Science, **2011, 5, 59-68**
Published Online November 2011 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijitcs.2011.05.08

An Optimization Model and DPSO-EDA for Document Summarization

Rasim M. Alguliev
Institute of Information Technology of Azerbaijan National Academy of Sciences
E-mail: rasim@science.az

Ramiz M. Aliguliyev
Institute of Information Technology of Azerbaijan National Academy of Sciences
E-mail: r.aliguliyev@gmail.com

Chingiz A. Mehdiyev
Institute of Information Technology of Azerbaijan National Academy of Sciences
E-mail: depart13@iit.ab.az

***Abstract* – We model document summarization as a nonlinear 0-1 programming problem where an objective function is defined as Heronian mean of the objective functions enforcing the coverage and diversity. The proposed model implemented on a multi-document summarization task. Experiments on DUC2001 and DUC2002 datasets showed that the proposed model outperforms the other summarization methods.**

***Index Terms* – generic summarization; optimization model; balancing coverage and diversity; Heronian mean; discrete particle swarm optimization; estimation of distribution algorithm**

1. Introduction

With the exponentially growing of the Internet technology a huge amount of electronic documents are available online. It is difficult to identify the relevant information to satisfy the information needs of users. This explosion of electronic documents has made it difficult for users to extract useful information from them, and the user due to the large amount of information does not read many relevant and interesting documents. Text summarization techniques have been found to be effective with regard to helping users find relevant information faster. That is why the necessity of tools that automatically generate summaries arises. These tools are not just for professionals who need to find the information in a short time but also for large search engines such as Google, Yahoo!, AltaVista, and others, which could obtain many benefits in its results if they use automatic generated summaries. After that, the user only will require the interested documents, reducing the flow information.

The effectiveness and efficiency of a user's performance in an information-seeking task can greatly be improved if he/she needs to only look at a summary that includes the relevant information presented in his/her preferred manner. On the other hand, if the main idea is misrepresented and/or omitted altogether from a summary, it may take users more time to solve a target problem or, even worse, lead users to make incorrect decisions. Therefore, there is an important need to design a personalized text summarization system that takes into account both what a user is currently interested in and how a user perceives information. The latter factor is referred to as a user's cognitive styles. Paper [1] aims at studying the impact of a user's cognitive styles when assessing multi-document summaries. In particular, authors choose two dimensions of a user's cognitive style – the analytic and verbal dimensions – and study their impacts on how a user assesses a summary that was generated from a set of documents.

Document summaries can be classified into different types according to different dimensions. For example, a summary can be either *generic summary* or *query-relevant summary* (sometimes called *query-biased summary*) [2–5]. A query-relevant summary is biased towards a given query or topic, and a generic summary is produced without any additional clues and prior knowledge. Furthermore, a summary can be either *abstraction-based* or *extraction-based*. An extraction-based summary involves merely selecting sentences or text segments from the source document, while an abstraction-based summary involves paraphrasing sections of the source document. In general, an abstraction-based summary can condense a text more strongly than an extraction-based summary, but it is harder than extraction-based summary because it requires the use of natural language generation technology. Depending on the number of documents to be summarized, the summary also can be a single-document or a multi-document [6]. Single-document

Manuscript received September 21, 2011; revised November 5, 2011; accepted October 28, 2011

summarization can only condense one document into a shorter representation, whereas multi-document summarization can condense a set of documents into a summary. Multi-document summarization can be considered as an extension of single-document summarization and used for precisely describing the information contained in a cluster of documents and facilitate users to understand the document cluster. Since it combines and integrates the information across documents, it performs knowledge synthesis and knowledge discovery, and can be used for knowledge acquisition.

Document summarization methods can be broadly divided into extractive and abstractive. Extractive methods work by selecting a subset of existing words, phrases, or sentences in the original text to form the summary. In contrast, abstractive methods build an internal semantic representation and then use natural language generation techniques to create a summary that is closer to what a human might generate. Such a summary might contain words not explicitly present in the original. Extraction methods merely copy the information deemed most important by the system to the summary (for example, key clauses, sentences or paragraphs), while abstraction involves paraphrasing sections of the source document. In general, abstraction can condense a text more strongly than extraction, but the programs that can do this are harder to develop as they require the use of natural language generation technology, which itself is a growing field. In fact, majority of researches have been focused on summary extraction. In contrast to extractive techniques, abstractive techniques are more complicated to implement because they require extensive domain knowledge to interpret source texts and generate new ones [2, 3, 7].

Extractive document summarization clearly entails selecting the most salient information and putting it together in a coherent summary. The summary consists of multiple separately extracted sentences from document(s). Obviously, each of the selected sentences should individually be important. However, when many of the competing sentences are included in the summary, the issue of information overlap between parts of the output comes up, and a mechanism for addressing redundancy is needed. Therefore, when many of the competing sentences are available, given summary length limit, the strategy of selecting best summary rather than selecting best sentences becomes evidently important. Selecting the best summary is a global optimization problem in comparison with the procedure of selecting the best sentences. In addition, it is known that coverage and diversity are two main criteria that decide the quality of summary. In this paper, we propose a new document summarization model via sentence extraction to simultaneously deal with these two concerns during sentence selection. In present paper, document summarization modeled as a 0-1 programming problem where an objective function is defined as Heronian mean of the objective functions enforcing coverage and diversity. This model does not only pick sentences from collection with highest significant and takes into account overlap information between selected sentences. The model employs two levels of analysis: first level, every sentence is scored according to the features it covers and second level, when, before being added to the final summary, the sentences deemed to be important are compared to each other and only those that are not too similar to other candidates are included in the final summary. To solve the optimization problem has been used Discrete Particle Swarm Optimization based on Estimation of Distribution Algorithm (DPSO-EDA). The experimental results suggest that DPSO-EDA is a very promising algorithm for general-purpose. The performance of the proposed model is tested on DUC2001 and DUC2002 data sets and is compared with baseline systems. The effectiveness of the proposed approach is demonstrated.

The remaining of the paper is organized as follows. Section 2 gives brief review of the document summarization methods. Section 3 presents the proposed generic document summarization model. Section 4 describes DPSO-EDA. Section 5 gives some experimental results to show the performance of the proposed model. Finally, some conclusions are presented in Section 6.

2. Related work

Many document summarization methods have been proposed in literature. As mentioned earlier, the methods for document summarization can be either extraction-based or abstraction-based. Extraction-based methods usually involve assigning a saliency score to each sentence and then ranking the sentences in the document [2, 5, 7]. This section focuses on extraction-based methods.

Most recently, the graph-based models have been successfully applied for multi-document summarization. The models first construct a directed or undirected graph to reflect the relationships between the sentences and then apply the graph-based ranking algorithms PageRank and HITS to compute the rank scores for the sentences. The sentences with large rank scores are chosen into the summary. The work [8] proposes a document-based graph model (denoted as DGM) to explore document impact on the graph-based summarization, by incorporating both the document-level information and the sentence-to-document relationship in the graph-based ranking process. Wan and Yang [9] proposed two models to incorporate the cluster-level information into the process of sentence ranking. The first model is the Cluster-based Conditional Markov Random Walk Model (ClusterCMRW), which incorporates the cluster-level information into the link graph. The second model is the Cluster-based HITS Model (ClusterHITS), which considers the clusters and sentences as hubs and authorities in the HITS algorithm. In [10], the intertopic information is used for transferring word importance learned from known topics to unknown topics under a

learning-based summarization framework. To mine this information, the authors model the topic relationship by clustering all the words in both known and unknown topics according to various kinds of word conceptual labels, which indicate the roles of the words in the topic. Based on the mined relationships, the authors develop a probabilistic model using manually generated summaries provided for known topics to predict ranking scores for sentences in unknown topics.

When dealing with multi-document summarization, existing sentence ranking algorithms often assemble a set of documents into one large file. The document dimension is ignored. Wei et al. [11] develop two alternative models to integrate the document dimension into existing sentence ranking algorithms. They are the one-layer (i.e. sentence layer) document-sensitive model and the two-layer (i.e. document and sentence layers) mutual reinforcement model.

In [12], the proposed algorithm can adequately summarize professional documents that include plural sentences having high similarity to the query. This algorithm has two steps: first is specifying the conformity part for the summary from the document, second is generating the summary with easy understanding based on the part. In the algorithm, the sentence extraction is performed by a paragraph in the document. In [13], the best summary is defined to be the one, which has the minimum information distance to the entire document set. The best update summary has the minimum conditional information distance to a document cluster given that a prior document cluster has already been read.

Clustering-based summarization methods usually perform various clustering techniques on the term-sentence matrices formed from the documents. After the sentences are grouped into different clusters, a centroid score is assigned to each sentence based on the average cosine (or other) similarity between the sentence and the rest of the sentences in the same cluster. Finally, the sentences with the highest scores in each cluster are selected to form the summary [14–19].

In recent years, the optimization-based methods have been proposed for document summarization. Filatova and Hatzivassiloglou [20] modeled extractive text summarization as a maximum coverage problem that aims at covering as many conceptual units as possible by selecting some sentences. McDonald [21] formalized text summarization as a knapsack problem and obtained the global solution and its approximate solutions. Takamura and Okumura [22] represented text summarization as maximum coverage problem with knapsack constraint. Cheung et al. [23] proposed a formal optimization-based method for summarization content selection based on the p-median clustering paradigm, in which content selection is viewed as selecting clusters of related information. The work [24] introduces an optimal formulation for the widely used greedy maximum marginal relevance (MMR) algorithm. The optimization problems are formulated as an integer linear programming.

3. Mathematical formulation of document summarization

3.1. The cosine similarity

Given a document collection $\mathrm{D} = \{d_1, d_2, ..., d_N\}$, where N is the number of documents. For simplicity, the document collection is represented as a set of all sentences from all the documents in the collection, i.e. $\mathrm{S} = \{s_1, s_2, ..., s_n\}$, where s_i denotes i th sentence in D, n is the number of sentences in the document collection. We first introduce a similarity measure.

Let $T = \{t_1, t_2, ..., t_m\}$ represents all the terms in D, where m is the number of different terms. The cosine measure is the most popular measure for evaluating text similarity based on the vector space model. In this model each sentence s_i is located as a point in a m dimensional vector space, $s_i = [w_{i1}, w_{i2}, ..., w_{im}]$. The weight w_{ij} associated with term t_j in sentence s_i is calculated by the scheme *tf-isf*. This scheme combines the definitions of term frequency and inverse sentence frequency, to produce a composite weight for each term in each sentence. This weighting scheme assigns to term a weight (a weight to a term) in sentence given by

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

where tf_{ik} is the term frequency (i.e. denotes how many term t_k occurs in sentence s_i), n_k denotes the number of sentences in which term t_k appears. The term $\log(n / n_k)$, which is very often referred to as the *isf* factor, accounts for the global weighting of term t_k. The *isf* factor has been introduced to improve the discriminating power of terms in the traditional information retrieval.

The cosine similarity between sentences $s_i = [w_{i1}, w_{i2}, ..., w_{im}]$ and $s_j = [w_{j1}, w_{j2}, ..., w_{jm}]$ can be calculated as:

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

3.2. Optimization model

In multi-document summarization, diversity is a particularly important issue since sentences from different documents might convey the same information. A high quality summary should not only be informative about the remainder but also be compact. Sentence selection is the most important step for processing sentences as this step actually adds sentences to the summary. Its consideration in multi-document summarization should not be based solely on having the highest relevance score, but also on sentence length and redundancy removal. This section generates a summary by selecting salient sentences in given documents taking into account overlap information between selected sentences.

Objective for enforcing coverage. This objective attempts to find a subset of the sentences $D = \{s_1, s_2, ..., s_n\}$ that covers the main content of the document collection. Let $S \in D$ be the set of sentences constituting a summary, then we would like to maximize the similarity $sim(D,S)$ between the document collection and the summary. A major aspect identifying relevant information is to find out what a document about. A document will generally contain a variety of information centered on a main theme, and covering different aspects of the main topic. Coverage means that the generated summary should cover all subtopics as much as possible. Poor subtopic coverage is usually manifested by absence of some summary sentences. The following objective function is introduced to enforce coverage:

$$f_{\text{cover}}(X) = sim(O, O^S) + \sum_{i=1}^{n} sim(O, s_i) x_i \ . \qquad (3)$$

Here O and O^S denote the centers of the collection $D = \{s_1, s_2, ..., s_n\}$ and the summary $S = \bigcup_{i=1}^{n} s_i x_i$, respectively, where x_i denotes a binary variable of the presence of sentence s_i in the summary and $\bigcup$ is the concatenation operation. Sentence concatenation is the operation of joining the sentences end-to-end. Higher value of $f_{\text{cover}}(\cdot)$ corresponds to higher content coverage of summary.

k th coordinate o_k of the mean vector O is calculated as:

$$o_k = \frac{1}{n} \sum_{i=1}^{n} w_{ik} \ , \qquad (4)$$

and k th coordinate o_k^S of the mean vector O^S we define as:

$$o_k^S = \frac{1}{n_S} \sum_{s_i \in S} w_{ik} \ , \qquad (5)$$

where n_S denotes the number of sentences in the summary S and $k = 1, ..., m$.

Objective for enforcing diversity. When generating a summary, we also need to deal with the problem of repetition of information. This problem is especially important for multi-document summarization, where multiple documents will discuss the same topic.

Diversity argues that a summary should not contain similar sentences. In other words, sentences in a summary should have little overlap with one another in order to increase diversity. The following objective function models diversity:

$$f_{\text{diver}} = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} (1 - sim(s_i, s_j)) x_i x_j \ . \qquad (6)$$

Higher value of $f_{\text{diver}}(\cdot)$ corresponds to lower overlap in content between sentences s_i and s_j, i.e. higher value of objective (6) provides high diversity in the summary.

Constructing a single aggregate objective function. In general, in a multiobjective optimization problem, it is not possible to find a single solution that optimizes all objectives simultaneously. Therefore, one is interested to explore a set of solutions, called the Pareto optimal set, which are not dominated by any other solution in the feasible set. The corresponding objective vectors of these Pareto optimal points, named efficient points, form the Pareto front on the objective space. The most traditional approach to solve a multiobjective optimization problem is to aggregate the objectives (3) and (6) into a single objective. In this study, to aggregate the objectives f_{cover} and f_{diver} into a single objective the Heronian mean is used. Thus, we have the following model:

maximize

$$f = \frac{2}{3} \cdot \frac{f_{\text{cover}} + f_{\text{diver}}}{2} + \frac{1}{3} \sqrt{f_{\text{cover}} \cdot f_{\text{diver}}} \qquad (7)$$

subject to

$$\sum_{i=1}^{n} l_i x_i \le L \ , \qquad (8)$$

$$x_i \in \{0,1\} \ , \ \forall i \ , \qquad (9)$$

where L is the length of summary, and l_i is the length of sentence s_i.

Since the model (7)-(9) is an NP-hard problem, it cannot generally be solved in polynomial time. Therefore, to solve the problem (7)-(9) we utilized the DPSO-EDA proposed in [25].

In the following section, firstly, the PSO and EDA are briefly reviewed, and then the DPSO-EDA is described [25].

4. DPSO-EDA

The idea behind the PSO is to learn from individual's own experience and the best individual experience in the whole swarm. Among the existing metaheuristic algorithms, a well-known branch is the PSO which is a stochastic search procedure based on observations of social behaviors of animals, such as bird flocking and fish schooling. PSO has some advantages, such as parallel processing, good robustness and high computational efficiency, and these features have been successfully applied to a variety of complex optimization problems [26].

PSO is a swarm intelligence-based optimization technique, inspired by the social behavior of bird flocking, in which a swarm of some particles/birds gradually improve their movement towards a piece of food. Unlike many other metaheuristics, such as genetic algorithm and differential evolution, which use random solutions for generating new solutions, PSO uses some previously explored best particles (solutions) for improving a particle in question. This characteristic of PSO increases its probability for generating good particles, and hence, to converge faster to the optimum. At any instant of PSO, a particle changes its velocity $V_i(t) = [v_{i1}(t), ..., v_{in}(t)]$ and position $X_i(t) = [x_{i1}(t), ..., x_{in}(t)]$ by exploiting the best position $P_i^{best} = [p_{i1}^{best}, ..., p_{in}^{best}]$ attained so far by itself (personal best) or by any other particle of the swarm

$G^{best} = [g_1^{best},..., g_n^{best}]$ (global best). The mathematical formulation of PSO for a swarm of N_{sw} particles can be expressed as below [26]:

$$v_{ij}(t+1) = \omega \cdot v_{ij}(t) + c_1 r_1 (p_{ij}^{best}(t) - x_{ij}(t)) + c_2 r_2 (g_j^{best}(t) - x_{ij}(t)), \quad (10)$$

$$x_{ij}(t+1) = x_{ij}(t) + v_{ij}(t+1), \quad (11)$$

where ω is the inertia factor that weights the previous particle's velocity, $t = 1,2,...,$ indicates the iteration number. c_1 is the cognition learning factor and c_2 is the social learning factor; r_1 and r_2 are two independent random numbers uniformly distributed within the interval [0,1] and the values of r_1 and r_2 are not the same for every iteration.

The constants c_1 and c_2 are used to weight the velocity towards the particle's personal best, $(p_{ij}^{best}(t) - x_{ij}(t))$, and the velocity towards the global best solution, $(g_j^{best}(t) - x_{ij}(t))$, found so far by the whole swarm.

The personal best position of particle x_i at iteration $(t+1)$ is calculated as:

$$p_i^{best}(t+1) = \begin{cases} p_i^{best}(t) & \text{if } f(x_i(t+1)) \le f(p_i^{best}(t)) \\ x_i(t+1) & \text{if } f(x_i(t+1)) > f(p_i^{best}(t)) \end{cases} \quad (12)$$

where $f(\cdot)$ is the objective function (7) to be maximized.

At the iteration t the best position of swarm is computed as:

$$g^{best}(t) = \max\{f(x_1(t)),..., f(x_{N_{swarm}}(t))\}. \quad (13)$$

The position vector $p_i^{best}(t)$ (12) represents the best position of particle x_i till the t th iteration and $g^{best}(t)$ (13) represents the best position in the swarm at the t th iteration.

The coefficients c_1 and c_2 are recommended to keep the following relationship: $c_1 + c_2 \le 4$, where c_1 is equal to c_2 and ranges from $[0,4]$. A usual choice for the acceleration coefficients c_1 and c_2 is $c_1 = c_2 = 1.494$ [26]. However, other settings were also used in different papers. Ratnaweera et al. [27] introduced time-varying acceleration coefficients, which reduces the *cognitive* component, c_1 and increases the *social* component, c_2 of acceleration coefficient with-time. With a large value of c_1 and a small value of c_2 at the beginning, particles are allowed to move around the search space, instead of moving toward *personal best*. A small value c_1 and a large value of c_2 allow the particles converge to the global optima in the latter part of the optimization. The time-varying acceleration coefficients are given in Eqs. (14) and (15):

$$c_1 = (c_{1f} - c_{1i}) \cdot \frac{t}{t_{\max}} + c_{1i}, \quad (14)$$

$$c_2 = (c_{2f} - c_{2i}) \cdot \frac{t}{t_{\max}} + c_{2i}, \quad (15)$$

where c_{1i} and c_{2i} are the initial values of the acceleration coefficients c_1 and c_2, and c_{1f} and c_{2f} are the final values of the acceleration coefficients c_1 and c_2, respectively. The objective of this modification is to boost the global search over the entire search space during the early part of the optimization and to encourage the particles to converge the global optima at the end search. An improved optimum solution observes when c_1 decreases from 2.5 to 0.5 whereas c_2 increases from 0.5 to 2.5 over the full range of the search.

In the PSO, the inertia weight in (10) linearly decreases during the search iteration by (16):

$$\omega = (\omega_{\max} - \omega_{\min}) \frac{(t_{\max} - t)}{t_{\max}} + \omega_{\min}, \quad (16)$$

where $\omega_{\max}$ and $\omega_{\min}$ represent the higher and lower inertia weight values to control the inertia, respectively. The values of w decrease from $\omega_{\max}$ to $\omega_{\min}$. t is the current iteration and $t_{\max}$ is the maximum number of iterations. Through empirical studies, it has been observed that the optimal solution can be improved by varying the value from 0.9 at the beginning of the search to 0.4 at the end of the search for most problems.

Population size plays an important role in evolutionary methods. Robustness and computation cost of the algorithm are also affected by it. Small population size may result in local convergence; large size will increase the computational efforts and may make slow convergence. Thus, an appropriate population size can maintain the effectiveness of the algorithm. It is quite a common practice in the PSO literature to limit the number of particles to the range from 20 to 60 [26].

Most versions of PSO have operated in continuous and real number space. In continuous versions of PSO, described by Eqs. (10) and (11), velocity updating equation (Eq. (10)) consists of three parts. The first is previous velocity of the particle, the second and third parts are the terms associated with their best solutions in the past including personal best solution and global best solution. For a discrete problem expressed in a binary notation, a particle moves in a search space restricted to 0 or 1 on each dimension, and thus updating a particle represents changes of a bit that should be in either state 1 or 0.

Kennedy and Eberhart [28] developed a discrete PSO (DPSO) to solve combinatorial optimization problems. In the discrete version, a particle moves in a state space restricted to 0 and 1 on each dimension, where $v_{ij}(t)$ represents the probability of bit $x_{ij}(t)$ taking the value 1. Thus, the step for updating $v_{ij}(t)$ remains unchanged as shown in Eq. (10), except $p_{ij}^{best}(t)$ and $g_j^{best}(t)$ are integer numbers in $\{0,1\}$ in binary case. The resulted changes in position are defined as follows:

$$x_{ij}(t+1)=\begin{cases}1, \text{ if rand}_j < pr(v_{ij}(t+1))\\ 0, \text{ otherwise}\end{cases} \quad (17)$$

where $0 \le \text{rand}_j \le 1$ is a uniform random number.

In (17), the velocity value is constrained to the interval [0,1] using the following sigmoid function:

$$pr(v_{ij}(t+1))=\frac{1}{1+\exp(-v_{ij}(t+1))}, \quad (18)$$

where $pr(v_{ij}(t))$ denotes the probability of bit $x_{ij}(t)$ taking 1.

To avoid $pr(v_{ij}(t))$ approaching 1 or 0, a constant $v_{\max}$ as a maximum velocity is used to limit the range of $v_{ij}(t)$, that is, $v_{ij}(t) \in [-v_{\max}, v_{\max}]$.

More details about PSO can be found in [26, 29], which are comprehensive reviews published more recently.

To extend PSO to solve combinatorial optimization problems, Wang et al. [25] proposed a discrete PSO based on EDA (DPSO-EDA), which combines the information sharing mechanism of PSO and the idea of EDA.

Estimation of distribution algorithms (EDAs) are nature-inspired optimization methods, which guide the search process by estimating a probability distribution of the high quality individuals. The individuals encode possible solutions for the optimization problem. Instead of operating on individuals, EDAs are population-based stochastic heuristics. They replace the recombination and mutation in the standard genetic algorithms, with the estimation of a joint probability model. This new model can be used to generate new individuals.

In the DPSO-EDA, let $X_i(t)=[x_{i1}(t),...,x_{in}(t)]$, $x_{ij}(t) \in \{0,1\}$, be particle i with n bits at iteration t. The DPSO-EDA for the j th bit of particle i is described as follows:

$$x_{ij}(t+1)=w(0,w_1)\times \text{mutate}(x_{ij}(t))+w(w_1,w_2)\times \text{mutate}(EDA_{ij}(t))+w(w_2,1)\times \text{mutate}(g_j^{best}(t)), \quad (19)$$

where $0<w_1<w_2<1$, $w(\cdot,\cdot)$ and $\text{mutate}(\cdot)$ are a threshold function and a mutation or bit flipping function, respectively, and they are defined as follows:

$$w(a,b)=\begin{cases}1 & \text{if } a \le \text{rand}() < b\\ 0 & \text{otherwise}\end{cases} \quad (20)$$

$$\text{mutate}(y)=\begin{cases}1-y & \text{if rand}() \le pm\\ y & \text{otherwise}\end{cases}. \quad (21)$$

Thus, only one of the three terms on the right hand side of Eq. (19) will remain dependent on the current random number produced by function $\text{rand}()$, $\text{mutate}(y)$ mutates the binary bit y with a small mutation probability pm.

In order to keep the diversities in particle swarm, a mutation operator is also incorporated into the proposed algorithm. After each bit is decided in accordance with estimated marginal distribution, the mutation operator independently flips the bit of an individual with a mutation probability.

Then some explanations about the three terms on the right hand side of Eq. (19) are given. The first term is to keep the previous state of particle, and the third term is to learn from global best solution. In the second term, the idea of EDA is incorporated, and therefore a particle can learn from the global statistical information collected by the personal best experiences of all the particles.

The probability vector in the proposed algorithm can be learned and updated at each iteration for modeling the distribution of promising solutions. The DPSO-EDA uses a probability vector $P=(p_1,...,p_j,...,p_n)$ to characterize the distribution of promising solutions in the search space, where p_j is the probability that the value of the j th position of a promising solution is 1. New offspring solutions are thus generated by sampling the updated solution distribution model. The second term on the right hand side of Eq. (19) is determined by the probability vector $P=(p_1,...,p_j,...,p_n)$ in the following way:

$$EDA_{ij}(t))=\begin{cases}1 & \text{if rand}() < p_j\\ 0 & \text{otherwise}\end{cases}. \quad (22)$$

In the sampling process above, a bit is sampled from the probability vector P randomly.

The probability vector P is initialized by the following rule:

$$p_j=\frac{\sum_{i=1}^{N_{sw}} p_{ij}^{best}}{N_{sw}}. \quad (23)$$

p_j is the percentage of the binary strings with the value of the j th element being 1. p_j can also be regarded as the center of the personal best solutions of all the particles.

The probability vector in the DPSO-EDA can be learned and updated at each iteration for modeling the distribution of promising solutions. Since some elements of the offspring are sampled from the probability vector P, it can be expected that the offspring should fall in or close to a promising area. The sampling mechanism can also provide diversity for the search afterwards. At each iteration in the DPSO-EDA, the personal best solutions of all the particles are selected and used for updating the probability vector P. Therefore, the probability vector P can be updated in the same way as in the population-based incremental learning algorithm (PBIL) [30]:

$$p_j=(1-\lambda)p_j+\lambda\frac{\sum_{i=1}^{N_{sw}} p_{ij}^{best}}{N_{sw}}, \quad (24)$$

where $\lambda \in (0,1]$ is the learning rate. As in the PBIL, the probability vector P is used to generate the next set of sample points.

The learning rate λ balances the contributions between the old statistical information extracted from personal best solutions of the historical particles and the information of the personal best solutions of the current particles to the new probability vector. The bigger λ is,

the greater the contribution of personal best solutions of the current particles is. The setting of the learning rate has a direct impact on the trade-off between exploration and exploitation ability. For example, if the learning rate is 0 , there is no exploitation of the information gained through search. As the learning rate is increased, the amount of exploitation increases, and the exploration ability to search large portions of the problem space diminishes.

5. Experiments

In this section, the data set, evaluation metrics, the parameter setting of the DPSO-EDA are described, and then the experiment results for the proposed method are given.

5.1 Data Set

Generic multi-document summarization has been one of the fundamental tasks in DUC2001 [31] and DUC2002 [32] (i.e. task 2 in DUC2001 and task 2 in DUC2002), and we used the two tasks for evaluation. The two tasks aimed to evaluate generic summaries with a length of approximately 100 words or less. DUC2001 provided 309 news articles collected from TREC-9, and DUC2002 provided 567 English news articles collected from TREC-9 for the multi-document summarization task. The DUC2001 documents could be categorized into 30 news topics and the DUC2002 documents could be categorized into 59 news topics. Table 1 gives a short summary of the datasets. The sentences in each article have been separated and the sentence information has been stored files.

Table 1: Summary of datasets

	DUC2001	DUC2002
Task	Task 2	Task 2
Number of documents	309	567
Number of clusters	30	59
Data source	TREC-9	TREC-9
Summary length	100 words	100 words

All the documents were segmented into sentences using a script distributed by DUC. For the similarity calculation between sentences, the stopwords were removed and the remaining words were stemmed using Porter's stemmer [33]. For removing the stopwords, the stoplist from [34] was used.

5.2. Evaluation metrics

For evaluation, the ROUGE (Recall-Oriented Understudy for Gisting Evaluation) toolkit is used. ROUGE package is introduced by Lin and Hovy [35], which proposed that the summarization system can be evaluated by the unigram co-occurrence with human judges. ROUGE was adopted by the DUC conference only from 2004 onwards. It includes measures, ROUGE-N, ROUGE-L, ROUGE-W, ROUGE-S and ROUGE-SU, which automatically determine the quality of a machine summary by comparing it to other (ideal) summaries created by humans. These measures evaluate the quality of the summarization by counting the number of overlapping units, such N-grams, between the generated summary by a method and a set of reference summaries.

ROUGE-N measure compares N-grams of two summaries, and counts the number of matches. This measure is computed as [35]:

$$\text{ROUGE-N}=\frac{\sum_{S\in Summ_{ref}}\sum_{\text{N-gram}\in S}Count_{match}(\text{N-gram})}{\sum_{S\in Summ_{ref}}\sum_{\text{N-gram}\in S}Count(\text{N-gram})}, \quad (25)$$

where N stands for the length of the N-gram, $Count_{match}(\text{N-gram})$ is the maximum number of N-grams co-occurring in candidate summary and the set of reference-summaries. $Count(\text{N-gram})$ is the number of N-grams in the reference summaries.

For evaluation of the method, two of the ROUGE metrics in the experimental results is used: ROUGE-1 (unigram-based), ROUGE-2 (bigram-based). ROUGE-1 and ROUGE-2 compare the unigram and bigram overlap between the system summary and the manual summaries created by human, respectively.

5.3. Parameter setting of the DPSO-EDA

The parameters of the DPSO-EDA are set as follows: the swarm size, $N_{sw}=50$; the number of iteration, $t_{\max}=500$; the cognitive and social parameters, $c_{1i}=c_{2f}=2.5$, $c_{1f}=c_{2i}=0.5$, and the inertia weight, $w_{\max}=0.9$, $w_{\min}=0.4$. The constant $v_{\max}$ is set to 5 for all the particles. The optimization procedure used here is stochastic in nature. Hence, for each criterion function it has been run several times. The results reported in experiments are averages over 20 runs for each method.

In the DPSO-EDA, there are also several parameters to be selected: w_1 , w_2 , mutation rate pm and learning rate λ . In the DPSO-EDA, the value of w_1 is dynamically tuned from 0.4 to 0 according to the number of generations such that more exploration search is pursued during the early generations and the exploitation search is emphasized afterward:

$$w_1=(w_1^{\max}-w_1^{\min})\frac{(t_{\max}-t)}{t_{\max}}+w_1^{\min}, \quad (26)$$

where $w_1^{\max}=0.4$ and $w_1^{\min}=0$ represent the higher and lower values, respectively.

The value of w_2 determines the relative importance of p_i^{best} and g^{best} , therefore $w_2=0.2w_1+0.8$ is set. The bit mutation probability pm is set to a small value 0.001. The parameter $\lambda=0.1$ is also set.

All the simulations were implemented in Delphi 7 on a Pentium Dual CPU, 1.6 GHz PC with 512 KB cache and 1 GB of main memory in Windows XP environment.

5.4. Performance evaluation

The proposed method is compared with the following methods: DGM [8], ClusterHIST [9], ClusterWICER [19], and UnifiedRank [36]. All the simulations were implemented in Delphi 7 on a Server running Windows Vista with two Dual-Core Intel Xeon CPU 4 GHz and 4Gb memory.

Table 2: Comparison results on DUC2001

Methods	ROUGE-1	ROUGE-2
Our method	**0.3993**	**0.0832**
DGM	0.3735	0.0661
ClusterHITS	0.3742	0.0688
ClusterWICER	0.3814	0.0751
UnifiedRank	0.3636	0.0650

Table 3: Comparison results on DUC2002

Methods	ROUGE-1	ROUGE-2
Our method	**0.4172**	**0.1026**
DGM	0.3901	0.0877
ClusterHITS	0.3855	0.0865
ClusterWICER	0,3928	0,0903
UnifiedRank	0.3834	0.0786

The proposed method first run on DUC2001 dataset. Further, the experiment is extended on DUC2002 dataset. Tables 2 and 3 provide the ROUGE scores of the methods on DUC2001 and DUC2002 datasets, respectively. As seen among other methods the ClusterWICER shows the best results compared to DGM, ClusterHITS, and UnifiedRank methods.

For visually illustration of the comparison, we use Figures 1 and 2. We subtract the UnifiedRank score (the worst score) from the scores of all the other methods and add the number 0.01 so that the difference can be observed more clearly.

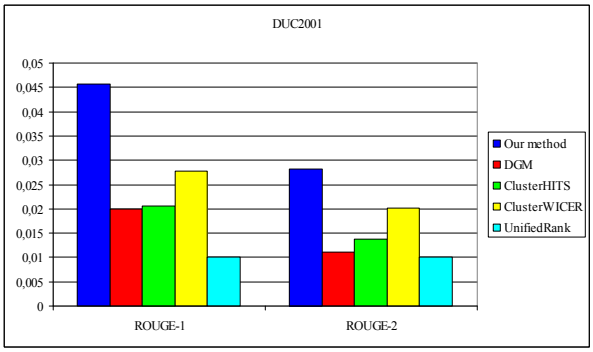

Figure 1. Comparison of the methods on DUC2001

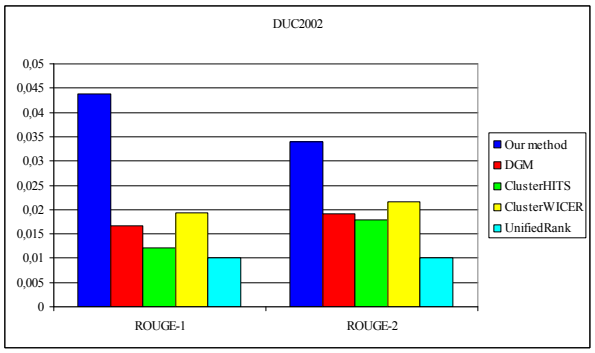

Figure 2. Comparison of the methods on DUC2002

Compared with the method ClusterWICER on DUC2001 (DUC2002) dataset the proposed method improves the performance by 4.69 (6.21)% and 10.79 (13.62)% in terms ROUGE-1 and ROUGE-2 metrics, respectively. For comparison, the relative improvement is used. The relative improvement is calculated as $(b-a)*100/a$ when b is compared to a.

6. Conclusion

In the paper, a novel text summarization model based on 0-1 non-linear programming problem is proposed. The proposed model covers main content of the given document(s) through sentence assignment. When comparing the proposed method to several existing summarization methods on DUC2001 and DUC2002 datasets, we found that the proposed method could improve the summarization results significantly. The methods were evaluated using ROUGE-1 and ROUGE-2 metrics.

References

[1] H. Nguyen, E. Santos, and J. Russell, Evaluation of the impact of user-cognitive styles on the assessment of text summarization, IEEE Transactions on Systems, Man, and Cybernetics Part A: Systems and Humans, 2011, 10.1109/TSMCA.2011.2116001.

[2] I. Mani and M.T. Maybury, Advances in automatic text summarization, MIT Press, Cambridge, 1999, 442p.

[3] Y. Ouyang, W. Li, S. Li, and Q. Lu, Applying regression models to query-focused multi-document summarization, Information Processing & Management, 2011, vol.47, no.2, pp.227–237.

[4] C.R. Chowdary, M. Sravanthi, and P.S. Kumar, A system for query specific coherent text multi-document summarization, International Journal on Artificial Intelligence Tools, 2010, vol.19, no.5, pp.597–626.

[5] X. Wan and J. Xiao, Exploiting neighborhood knowledge for single document summarization and keyphrase extraction, ACM Transactions on Information Systems, 2010, vol.28, no.2, Article 8, 34p.

[6] X. Wan, Using only cross-document relationships for both generic and topic-focused multi-document

summarizations, Information Retrieval, 2008, vol.11, no.1, pp.25□49.
[7] M. Kutlu, C. Cigir, and I. Cicekli, Generic text summarization for Turkish, The Computer Journal, 2010, vol.53, no.8, pp.1315-1323.
[8] X. Wan, An exploration of document impact on graph-based multi-document summarization, Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing, Honolulu, Hawaii, October 25-27, 2008, pp.755–762.
[9] X. Wan and J. Yang, Multi-document summarization using cluster-based link analysis, Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Singapore, July 20–24, 2008, pp.299-306.
[10] Y. Ouyang, W. Li, S. Li, and Q. Lu, Intertopic information mining for query-based summarization, Journal of the American Society for Information Science and Technology, 2010, vol.61, no.5, pp.1062–1072.
[11] F. Wei, W. Li, and Y. He, Document-aware graph models for query-oriented multi-document summarization, Studies in Computational Intelligence, 2011, vol.346, pp.655–678.
[12] C. Otani, M.K. Hoo, Y. Oda, T. Furue, Y. Uchida, and O. Yoshie, Query-biased summarization considering difference of paragraphs, IEEJ Transactions on Electronics, Information and Systems, 2010, vol.130, no.12, pp.2256-2265+21.
[13] 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, 2010, vol.25, no.4, pp.739–749.
[14] D. Wang and T. Li, Document update summarization using incremental hierarchical clustering, Proceedings of the ACM 19th Conference on Information and Knowledge Management, Toronto, Canada, October 26–30, 2010, pp.279–287.
[15] R.M. Alguliev and R.M. Aligulyiev, Automatic text documents summarization through sentences clustering, Journal of Automation and Information Sciences, 2008, vol.40, no.9, pp.53□63.
[16] R.M. Aliguliyev, A new sentence similarity measure and sentence based extractive technique for automatic text summarization, Expert Systems with Applications, 2009, vol.36, no.4, pp.7764□7772.
[17] R.M. Aliguliyev, Clustering techniques and discrete particle swarm optimization algorithm for multi-document summarization, Computational Intelligence, 2010, vol.26, no.4, pp.420-448.
[18] R.M. Alguliev and R.M. Aliguliyev Evolutionary algorithm for extractive text summarization, Intelligent Information Management, 2009, vol.1, no.2, pp.128–138.
[19] R.M. Aliguliyev, The two-stage unsupervised approach to multidocument summarization, Automatic Control and Computer Sciences, 2009, vol.43, no.5, pp.276–284.
[20] E. Filatova and V. Hatzivassiloglou, A formal model for information selection in multi-sentence text extraction, Proceedings of the 20th International Conference on Computational Linguistics, Geneva, Switzerland, August 23-27, 2004, pp.397-403.
[21] R. McDonald, A study of global inference algorithms in multi-document summarization, Proceedings of 29th European Conference on IR Research, Rome, Italy, April 2□5, 2007, Springer-Verlag, LNCS, 2007, no.4425, pp.557□564.
[22] H. Takamura and M. Okumura, Text summarization model based on maximum coverage problem and its variant, Proceedings of the 12th Conference of the European Chapter of the ACL, Athens, Greece, March 30 □ April 3, 2009, pp.781□789.
[23] J.C.K. Cheung, G. Carenini and R.T. Ng, Optimization-based content selection for opinion summarization, Proceedings of the 2009 Workshop on Language Generation and Summarization (ACL-IJCNLP), Singapore, 6 August 2009, pp.7–14.
[24] K. Riedhammer, B. Favre, and D. Hakkani-Tür, Long story short – global unsupervised models for keyphrase based meeting summarization, Speech Communication, 2010, vol.52, no.10, pp.801–815.
[25] J. Wang, Y. Cai, Y. Zhou, R. Wang, and C. Li, Discrete particle swarm optimization based on estimation of distribution for terminal assignment problems, Computers & Industrial Engineering, 2011, vol.60, no.4, pp.566–575.
[26] R. Poli, J. Kennedy, and T. Blackwell, Particle swarm optimization: an overview, Swarm Intelligence, 2007, vol.1, no.1, pp.33–57.
[27] A. Ratnaweera, S.K. Halgamuge, and H.C. Watson, Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients, IEEE Transactions on Evolutionary Computation, 2004, vol.8, no.3, pp.240–255.
[28] J. Kennedy and R. Eberhart, Particle swarm optimization, Proceedings of the IEEE International Conference on Neural Networks, Perth, Australia, 27 November – 01 December 1995, vol.4, pp.1942–1948.
[29] D. Martens, B. Baesens, and T. Fawcett, Editorial survey: swarm intelligence for data mining, Machine Learning, 2011, vol.82, no.1, pp.1–42.
[30] S. Baluja, Population-based incremental learning: a method for integrating genetic search based function optimization and competitive learning. School of Computer Science, Carnegie Mellon University, Pittsburgh, Technical Report CMU-CS-94-163, 41pp.
[31] DUC2001: http://www-nlpir.nist.gov/projects/duc/guidelines/2001.html
[32] DUC2002: http://www-nlpir.nist.gov/projects/duc/guidelines/2002.html

[33] Porter Stemming Algorithm: http://www.tartarus.org/martin/PorterStemmer/
[34] English stoplist: ftp://ftp.cs.cornell.edu/pub/smart/english.stop
[35] C.-Y. Lin and E. Hovy, Automatic evaluation of summaries using n-gram co-occurrence statistics, Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language, Edmonton, Canada, May 27 – June 1, 2003, vol.1, pp.71–78.
[36] X. Wan, Towards a unified approach to simultaneous single-document and multi-document summarizations, Proceedings of the 23rd International Conference on Computational Linguistics, Beijing, China, August 23-27, 2010, pp.1137–1145.

Rasim M. Alguliev: Dr.Sc. in computer science, Professor. Director of the Institute of Information Technology of Azerbaijan National Academy of Sciences. His research interest include information security of computer networks, information society, e-government, web mining, text mining, data mining, and evolutionary algorithms.

Ramiz M. Aliguliyev: Dr.Sc. in computer science. Head of department of the Institute of Information Technology of Azerbaijan National Academy of Sciences, interested in web mining, text mining, data mining, and evolutionary algorithms.

Chingiz A. Mehdiyev: PhD student in the Institute of Information Technology of Azerbaijan National Academy of Sciences, interested in document summarization and evolutionary algorithms.

