Computing and Informatics, Vol. 38, 2019, 1237–1271, doi: 10.31577/cai_2019_5_1237

PROBABILISTIC PAGE REPLACEMENT POLICY IN BUFFER CACHE MANAGEMENT FOR FLASH-BASED CLOUD DATABASES

Atul O. Thakare, Parag S. Deshpande

Department of Computer Science and Engineering
Visvesvaraya National Institute of Technology
South Ambazari Road, Nagpur, Maharashtra
India – 440010
e-mail: aothakare@gmail.com, psdeshpande@cse.vnit.ac.in

Abstract. In the fast evolution of storage systems, the newly emerged flash memory-based Solid State Drives (SSDs) are becoming an important part of the computer storage hierarchy. Amongst the several advantages of flash-based SSDs, high read performance, and low power consumption are of primary importance. Amongst its few disadvantages, its asymmetric I/O latencies for read, write and erase operations are the most crucial for overall performance. In this paper, we proposed two novel probabilistic adaptive algorithms that compute the future probability of reference based on recency, frequency, and periodicity of past page references. The page replacement is performed by considering the probability of reference of cached pages as well as asymmetric read-write-erase properties of flash devices. The experimental results show that our proposed method is successful in minimizing the performance overheads of flash-based systems as well as in maintaining the good hit ratio. The results also justify the utility of a genetic algorithm in maximizing the overall performance gains.

Keywords: Cloud databases, data mining, flash solid state drives, adaptive optimization, genetic algorithm

Mathematics Subject Classification 2010: 68Uxx

1 INTRODUCTION

1.1 Flash Based Systems

Most storage systems built on conventional hard disk drives (HDDs) suffer from technical limitations, such as low random access performance and high power consumption. The mechanical nature of HDDs prevents these problems to get addressed via technology evolution. Flash memory based Solid State Drive (SSD) is a type of electrically erasable and programmable read-only memory (EEPROM) which plays a crucial role in revolutionizing the storage system design. With the continuous decrease in price and increase in storage capacity, the usage of SSDs is growing in mobile devices, embedded computing and portable devices such as PDAs (personal digital assistants), digital audio players, digital cameras, HPCs (handheld PCs), etc. Due to its several other attractive features like high reliability (shock resistant), high I/O performance (fast random reads) and low power consumption it is also considered for replacing magnetic hard disks in enterprise database servers.

Unlike rotating hard disk drives which have mechanical movement overheads measured in terms of seek time and rotational latency, the flash SSD requires no movement of any of its mechanical part while read or write access to it. Unlike hard disks which have nearly uniform read-write cost, random writes to SSDs are much slower than random reads from it because of its erase-before-write limitation. In SSDs typically, write operations are about ten times slower than read operations, and erase operations are about ten times slower than write operations [13].

The storage space of flash memory is divided into fixed-sized blocks, each one containing fixed-sized pages. Hence block size (usually 16 KB) is multiple of page size (usually 512 bytes). Read/write operations are performed in units of pages whereas erase operation is performed in units of blocks. Flash memory has the typical property of writing a page only once, because of which page data cannot be updated in-place. Hence when some contents of a page need to be modified, the entire page must be written into a free page slot and the old page contents need to be invalidated (out-of-page updates) [25]. The invalidated page can be written only after erasing the block containing that page, i.e., a written memory must be erased before it can be written again. SSDs use an additional driver software called flash translation layer (FTL) which works in coordination with the host operating system, and whose function is to map the logical blocks to the physical pages on the flash device [27]. SSDs have limited processing power and random access memory (RAM), to manage relatively much larger sized flash memory. To achieve good performance with limited resources, FTLs are designed to hide the mismatch between the write and the erase operations, by exploiting the localities in the write requests. But in case there is a high percentage of random write requests, the performance of SSDs can drop significantly. Especially in cloud environments where the page access patterns are highly complicated due to numerous clients with diverse requirements accessing the cloud storage simultaneously, the random write performance of SSDs is extremely crucial.

The erase operation is performed by garbage collection policy, whenever a sufficient number of free slots are not available in the flash memory. Hence the increase in the number of writes will cause an increase in the number of erase operations, which are slowest among the 3 flash operations. Another unique feature of flash memory is limited erase count, i.e., a limited number of erase operations are allowed to be carried on each block. After a specified number of erase operations (in the range 10 000 to 1 000 000 depending on its physical characteristics) the block will become unreliable. Due to this, the lifetime of flash memory is shorter than that of devices like the hard disk.

Moreover, due to the electrical properties of flash cells, garbage collection overheads are higher in the case of random writes as compared to sequential writes. If the write requests are randomly distributed over the logical block address space, sooner or later all the physical blocks in flash memory will be fragmented, severely damaging the performance of garbage collection by increasing I/O latencies [15].

Based on the above facts we can conclude that, in flash SSDs, replacement of a dirty page can induce more cost than the cost of replacement of a clean page. This reduces the guarantees of optimizing I/O performance with the reduction in the miss rate. In other words the relationship between I/O cost and miss rate may not be consistent in the flash disks. Hence the effectiveness of buffer management schemes can be measured in terms of hit rate and average I/O service cost per page fault. The average I/O cost depends on flash memory characteristics as well as read-write patterns in the workload.

As flash memory is becoming a promising alternative to replace hard disks, like other software systems, the database systems also need to device techniques to cope with flash I/O properties to reduce I/O latencies. Unfortunately, most of the existing disk-oriented buffer replacement algorithms aim at better performance by considering

1. read and write operations equally, i.e., having the same latency,
2. hit ratio improvement as the only means of performance improvement,

due to which hard disk-based DBMS buffer managers are inefficient to deliver good DBMS performance on flash-based systems.

While designing the buffer management policy for flash-oriented systems, its asymmetric read-write feature needs to be taken into account in addition to the locality of pages in the main memory. Hence the buffer management schemes for flash memory need to aim at not only the better hit ratios but also to minimize the replacement costs incurring when a dirty page has to be propagated to flash memory to make room for a requested page currently not in buffer. In other words, the replacement policy should be able to minimize the number of erase operations by controlling the number of write operations on flash memory and, at the same time, avoid a significant fall in the hit ratio, because the fall in the hit ratio will lead to an additional increase in the number of read operations.

1.2 Cloud Databases

Cloud computing has emerged as an important computing paradigm which refers to both the applications delivered as services over the Internet and the hardware and systems software in the data centers that provide those services. The services offer facilities for data storage, data processing, and information management [26]. The services have been referred to as Software as a Service (SaaS), IaaS (Infrastructure as a Service) and PaaS (Platform as a Service). A cloud database is a database that typically runs on a cloud computing platform (private, public or hybrid), access to it is provided as a service.

Nowadays many applications are hosted on cloud databases where several applications share the same database instance. Such a database management system exhibits periodic behavior in terms of data references. For example, US customers access data at a particular time while Japanese customers access data at some other time. The periodicity of data references is translated into periodic page references. This periodicity of page references can be used as a new parameter to improve cache performance by improving page replacement policy.

1.3 Our Contributions

To meet the above-mentioned challenges on cloud-based flash memory devices, we propose two cost-based adaptive buffer-management algorithms based on the probabilistic model of page references, namely Accurate Probabilistic Adaptive Clean First Algorithm (APR-ACF) and Approximate Probabilistic Adaptive Clean First Algorithm (PR-ACF). Our algorithms focus on achieving the following objectives:

1. Reducing the number of write/erase operations by considering the read-write cost ratio of flash disks at each replacement.
2. Maintaining the hit rate as high as possible by considering the recency, frequency, and periodicity of page references.

An additional feature of our algorithms is that it is self-tuning to respond to changes in page reference patterns, by analyzing the frequency, recency, and periodicity of page references in an online as well as offline manner.

Summarizing:

- We have introduced the idea of the periodicity of page references considering recent trends in database applications, especially in a cloud environment.
- We have designed the concept of buffer management based on the probability distribution of page reference which is more generalized and is capable to map any existing specialized algorithms.
- The probability of reference is used to calculate which pages may be referenced soon. The probability of reference gives us the information about pages in the buffer cache, which one of them is a cold/warm/hot page.

- Our flash aware buffer management scheme prefers to make replacement in the COLD REGION; a part of the buffer cache which contains the pages with a low probability of reference.
- Within the COLD region it prefers to replace the CLEAN page with the lowest probability of reference. In the absence of a CLEAN page in the COLD region, it replaces the DIRTY page with the lowest probability of reference.
- Our algorithm is designed to retain in the buffer:
  1. The dirty pages with medium to high reference probability (warm/hot dirty pages).
  2. The clean pages with high reference probability (hot clean pages).
- Our algorithm is designed to quickly remove from buffer:
  1. The dirty pages with low reference probability (cold dirty pages).
  2. The clean pages with low to medium reference probability (cold/warm clean pages).

  If a conflict arises, clean pages are preferred over dirty pages for replacement. The hotness or coldness of a page is decided based on its current probability of reference. Here, the reference probability of a page is computed based on recent references and past historical references, as described in Section 3.
- Our algorithm adapts with the changing page reference patterns by dynamically resizing the COLD and HOT regions of the buffer cache.

The remainder of this article is structured as follows. The related works in this field of cache management in hard disks based systems and flash-based systems are introduced in Section 2 before describing the proposed buffer replacement algorithms APR-ACF and PR-ACF in Section 3. Section 4 describes a page access probability model and Section 5 explains the proposed page replacement algorithm using the probabilistic model described in Section 4. Section 6 presents cost-benefit analysis, whereas Section 7 explains a practically more efficient version of the page replacement algorithm described in Section 5. The detailed analysis of the simulation experiments and the results on various traces of different characteristics are explained in Section 8. Section 9 explains the need for finding the optimal set of time intervals, its application in maximizing the overall performance benefits, and the use of genetic algorithms in defining the optimal time intervals.

2 RELATED WORK

2.1 Literature Survey

Since the database management system is accessed by various types of users and data is stored in different types of storage structures, more complex referencing patterns are seen. Such page referencing patterns are categorized as sequential references, random references, hierarchical references and looping hierarchical references [1]. Such referencing patterns describe a new query behavior model, the query locality set model [QLSM] and based on it, a buffer management algorithm DBMIN was proposed by Chou and DeWitt [2]. Based on the looping behavior of the operations the hot set model is proposed by Sacco and Schkolnick [3]. The domain separation algorithm proposed by authors in [3] divides the buffer pool into several domains. Each domain represents a separate data structure like B-Tree or Cluster. So when B-Tree is accessed it is ensured that non leaf parent node always resides in memory. Another problem in database cache optimization is handling infrequent long queries. If the query is infrequent and requires lot of disk page access then to bring the required disk pages in the cache, whole existing buffers will be replaced. This will lead to removal of existing and stable working set in the cache and can result in lot of cache misses afterwards. Nowadays most of the database instances are residing in cloud environment and same database instance can be shared by multiple users across the globe. Each such user may not access whole data but some specific set of data. For example users from U.S. may access data of baseball products, while the users from Brazil may be interested only in football products. So if the data of such products is stored in clustered table and in data pages $\langle B_m \dots B_{m+i} \rangle$ [pages having data of baseball products] and $\langle B_n \dots B_{n+j} \rangle$ [pages having data of football products], then the pages, $\langle B_m \dots B_{m+i} \rangle$ may be referred more frequently in 12–18 UTC while pages $\langle B_n \dots B_{n+j} \rangle$ may be referred more frequently in 17–23 UTC. So the page referencing pattern shows certain periodicity and can be modeled using some probability distribution such as Gaussian distribution. Due to advent of superior hardware and advanced tracing systems in database, the detailed historical page trace is available indicating page number and page reference time which can be processed offline to construct probability distribution of page reference.

Nowadays different storage options like index organized tables, clusters, indexes, and partitions are available for storing data. In such structures, all data related with same key are stored in the same page or in pages that are physically contiguous to each other. Same query may generate different referencing pattern if the underlying storage structure is different. The heterogeneity of users in terms of their likings, their operating time, 24×7 application environment and different underlying storage structures has generated many complex reference patterns and conventional buffer management algorithms, which are based on one or two reference pattern(s), may not provide better solution for buffer management.

The efficient buffer management system demands accurate estimation of future probability of reference. The estimation should consider heterogeneity of users and their operating time, different storage structures and current pattern of references. Such estimation requires analysis of past references which normally provided by database management systems and decision of replacement should be taken using complete past pattern rather than using single event like LRU. Modern machine learning techniques can be used to estimate probability distribution and allows us to map the problem of replacement as classification problem. The only argument against such methodology is time required for such complex decision making which is very critical in cache management. Modern hardware can provide better implementation of such algorithms and the improvement in hit ratio, which may substantially reduce disk access, may outweigh cost incurred due to complex buffer management.

2.1.1 Traditional Buffer Replacement Algorithms

Traditional replacement algorithms, which are hard disk-oriented buffer management algorithms, primarily focus on the hit ratio for good performance. Many algorithms have been proposed so far, most of which are based on the recency and/or frequency property of page references. Among them, the best-known algorithms are LRU, CLOCK [17], LRU-2 [18], 2Q [21], LRFU [19], etc. Few more of them are algorithms like LIRS [20], ARC, CAR [15], CART, Clock-Pro [22], etc., with an additional adaptability, i.e., self-tuning feature.

2.1.2 Buffer Management Algorithms for Flash-Based DBMSs

Most of the asymmetry-aware flash buffering algorithms indicate two design points:

1. Distinguish clean pages (pages which contains the same copy of the original data in flash memory; need not be written back on flash memory at the time of replacement) and dirty pages (pages which are modified after reading the original data from flash memory into main memory; hence need to be written back on the flash memory whenever they are replaced).
2. Compare the locality of the two kinds of pages to make the replacement decision.

Most of these algorithms try to reduce the number of write operations, by delaying the process of evicting the dirty pages to enhance the I/O performance. One of the first flash-based buffering algorithm is proposed by Park et al. [7], namely CF-LRU which is based on the principle of holding the dirty pages as long as possible. For giving additional stay to dirty pages it enforces quicker replacement of clean pages. Li et al. [13] published the Cold-Clean-First LRU (CCF-LRU) algorithm which evict the cold clean pages first. It gives priority to replacing clean pages, what many times results in quick replacement of newly inserted pages, thus reducing the hit ratio. In absence of clean page, it prefers to replace cold dirty pages. Based on CCF-LRU, Jin et al. [8] designed Adaptive Double LRU (AD-LRU) algorithm to further improve the runtime efficiency. AD-LRU maintains the cold and hot pages in two separate LRU queues and dynamically adjusts their length according to the reference patterns. Jung et al. [16] propose the LRU-WSR algorithm which uses a Write Sequence Reordering (WSR) strategy. LRU-WSR prolongs the stay of hot dirty pages in the buffer and prefers replacing clean pages or cold-dirty pages. The authors On et al. [12] develop a FD-Buffer algorithm in which clean and dirty pages are separated into two pools. The size ratio of the two pools is dynamically adjusted based on the read write asymmetry property of the flash memory and the runtime workload. Prober scheme [10] is efficient to exploit the workloads which contain large sequential write sequences, especially in the write-dominant traces. To address the problem of cache pollution, Prober identifies large sequential write request at early stage and enforces its quick replacement by labeling it as a cold page. By monitoring I/O access patterns at runtime, Hystor [11] can effectively identify pages that can result in long latencies or are semantically critical (e.g. file system metadata), and stores them in SSDs for future accesses to achieve a significant performance improvement. To further enhance write performances, Hystor also serves a write-back buffer to speed up write requests.

3 PROPOSED WORK

3.1 Estimating Probability of Reference

In the proposed work we have provided a method for estimating the probability of reference of the page in any time interval and suggested buffer management algorithm based on it. We have also shown how different types of patterns can be accommodated in this model and how algorithms like LRU and LFU can be simulated by changing different algorithm parameters. We have also discussed the cost-benefit analysis and analyzed to find out under what conditions the current strategy provides benefit to cost ratio.

Moreover the proposed page referencing model essentially captures most of the page referencing behavior [24]. The proposed method is based on the following assumptions which are valid in most of the practical conditions.

1. The 24×7 internet-based database systems show periodicity of references due to different users across the globe.
2. The queries are made efficient by using various storage structures that store the related data physically adjacent to each other.
3. The query load generally comprises periodic frequent queries and infrequent queries.
4. Based on the estimated probability of references we can determine cold, warm and hot pages in the buffer cache.
5. Performance of flash-based database system can be optimized by modifying the buffer management policy to minimize the replacement of hot or warm dirty pages as well as hot clean pages if possible.

4 MODELING PAGE ACCESS PROBABILITY

The probability model [5] of page reference is based on recent references of the page and periodic references of the page which are captured from past data in a specific time interval. Generally in managing buffer cache, the page with a lower probability of reference is replaced and page with a higher probability of reference is kept in the buffer cache. The probability of future reference from current time τ_c to future time $\tau_c + \Theta$ is estimated using following principles:

1. If the page is referred frequently in the recent past, then its probability of reference in the future is higher. The probability estimated using the principle is denoted as R_{b_i,t_1,t_2} where b_i is page id and $t_1 - t_2$ ($t_2 > t_1$) indicates time interval.
2. If the page is referred frequently in some interval in the past, then there is higher probability of referencing it in future in the same interval. The probability estimated using this principle is denoted as H_{b_i,t_1,t_2}.
3. The total probability of reference T_{b_i,t_1,t_2} is calculated as weighted sum R_{b_i,t_1,t_2} and H_{b_i,t_1,t_2}. Thus $T_{b_i,t_1,t_2} = \omega_1 R_{b_i,t_1,t_2} + \omega_2 H_{b_i,t_1,t_2}$ where $\omega_2 + \omega_1 = 1, \omega_i \in$ Domain of real numbers between 0.0 and 1.0.

In calculating probability in all cases, probability distribution is estimated using Parzen window technique with Gaussian kernel [4]. This technique provides all the essential properties to capture behavior of page referencing pattern which is explained in the remaining part of this section.

4.1 Calculating Probability R_{b_i,t_1,t_2} Based on Recent References

Using Parzen window classifier with normal distribution [4], probability density estimation function of each page is estimated (Parzen window is used because the approximate estimation can be chosen by changing distribution parameters).

Let

- N be total number of page references in time period t.
- t be total time period of data collection specified in small time units. For example, if data is collected for 10 days and time unit is minutes then $T = 10 * 1\,440 = 14\,400$ where $1\,440$ is the number of minutes for one day, i.e., 24 hours.
- τ_i be time instance when page b is referred to in the specified interval.
- $S = \tau_1, \tau_2, \tau_3, \tau_4, \ldots, \tau_k$ indicate the set of time units where a page is referred. For example, if a page is referred at 11 am on day 1 and 11.05 am on day 2 then $S = \{660, (1\,440 + 665)\}$.
- P be periodicity of reference. For example, page referencing pattern is repeated for each day, then $P = 1\,440$, i.e., the number of time units in one day.
- σ be a user-defined parameter which controls the effect of past reference on probability. The higher value of σ indicates bigger effect of reference on the future pattern. For experimentation, value of σ is chosen as 1.

Hence, if page b_i is accessed at time $\tau_1, \tau_2, \ldots, \tau_k$ then probability density function of page b_i is

$$P_{bi} = \frac{1}{N} \sum_{i=1}^{k} \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{-1}{2}\left(\frac{t-\tau_i}{\sigma}\right)^2}. \tag{1}$$

$P_{bi} = 0$ if b_i is not referred in the past. Probability of page reference b_i in the interval t_1 to t_2 is

$$P_{b_i,t_1,t_2} = P_{b_i}(t_1 \leq t \leq t_2) = \int_{t_1}^{t_2} \left(\frac{1}{N} \sum_{i=1}^{k} \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{-1}{2}\left(\frac{t-\tau_i}{\sigma}\right)^2} \right) \mathrm{d}t. \tag{2}$$

The density function satisfies following essential properties of probability function:

$$P_{b_i}(t_1 \leq t \leq t_2) \neq 0 \quad \text{if } t_2 > t_1, \tag{3}$$

$$P_{b_i}(-\infty \leq t \leq \infty) = \frac{N_K}{N} \tag{4}$$

where N_K is total historical references of page b_i. If $B = b_1, b_2, \ldots, b_n$ is set of all pages then

$$P_{b_i}(-\infty \leq t \leq \infty) = \sum_{i=1}^{n} P_{b_i,-\infty,\infty} = \frac{1}{N} \sum_{i=1}^{n} N_k = 1. \tag{5}$$

4.2 Calculating Probability $\mathbf{H_{b_i,t_1,t_2}}$ Based on Past References of Historical Data

In this case, the time period is divided into fixed interval of equal size. For example if the time period is a day then it is divided into time intervals of one hour, so number of time intervals is 24 from 0–1 am, 1–2 am, etc. For practical implementation, they are represented using number of minutes from the start of period as 0–60, 61–120, 121–180, etc. If the page is referred many times in the past in some interval then it is also likely to be referred in the same interval in future. For each such interval working set of pages is calculated as set of top N pages in terms of probability of reference in that interval.

Here probability density function is similar to previous case but instead of using τ_1, τ_2 as absolute time, the time is calculated from start of the period. For example if page is referred on day 1 at 5:30 pm and on day 2 at 5:45 pm then τ_1 & τ_2 are taken as 1 050 and 1 065 minute, i.e. number of minutes from start of the period. The probability density function of reference of page b_i is indicated as

$$H_{b_i} = \frac{1}{N} \sum_{i=1}^{k} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(\frac{-1}{2} \left(\frac{\mid t - \tau_i \mid}{\sigma} \right)^2 \right) \tag{6}$$

where $\mid t - \tau_i \mid$ indicates time difference between t and τ_i considering circular scale. The probability of reference between time period t_1 and t_2 is

$$H(t_1 \leq t \leq t_2) = \int_{t_1}^{t_2} H_{b_i} \mathrm{d}t. \tag{7}$$

4.3 Finding Out LRU (Least Recently Used), MRU (Most Recently Used), MFU (Most Frequently Used) Buffers from Probability Model

The above probabilistic model of page reference is able to capture various types of buffers which may be used to implement existing conventional algorithms. Many of the replacement policies consider least recently used buffers (LRU), most recently used buffers (MRU), frequently used buffers (FUB) as criteria for replacement or stay in buffer cache. These buffers are determined using following formulas. The least recently used buffer (LRU) can be found out using Equation (8):

$$B_{LRU} = {}_{i}^{argmin} P_{b_i,t_c,\infty} \tag{8}$$

where $P_{b_i,t_c,\infty}$ is calculated based only on most recent reference of b_i. The most recently referred buffer (MRU) can be found out using Equation (9):

$$B_i = {}_{i}^{argmax} P_{b_i,t_c-\theta,t_c} \tag{9}$$

where $\sigma \gg \theta$.

The most frequently used buffer (MFU) can be found out using Equation (9) with the following constraints on the value of σ:

$$0 \ll \sigma < \theta.$$

5 PROPOSED ALGORITHMS

Based on the above probabilistic model of page reference the proposed algorithm is designed as follows:

- Divide the period into number of intervals T_0, T_1, T_2, ...
- For each interval calculate the historical working set based on SQL traces as given in Section 4.2 and Section 5.1.
- For each page in working set, calculate probability according to Equation (9). For fast access the information of page and its probability is kept in a hash table.

5.1 Selecting Pages for Working Set

For each time interval, pages are sorted in descending order based on their computed probability value in that interval and first k pages are chosen where k is the size of buffer cache. The pages are chosen only if their probability is higher than predefined threshold.

5.2 Assigning Ranks to Working Set Pages

Within each working set, pages are given ranks based on their probability values.

5.3 Buffer Cache Organization

- Maintain list of pages which are referred in past along with their time of reference. This list is used to calculate probability R_{b_i,t_1,t_2} according to equations given in Section 4.1. Since it is not possible to store all past references, the size of list is m times the size of buffer cache where $m > 1$.
- Maintain list of pages in cache with their total probability value. The list is sorted in descending order of their probability value. The page having lowest probability is at the one end of the list which is called as COLD END while page having highest probability is at another end which is called as HOT END. The pages at the COLD END and HOT END are referred as '*bc*' and '*bh*', respectively.
- The variable SPLIT divides the buffer cache into two parts which are HOT REGION and COLD REGION. Initially SPLIT is initialized to 50 % of the buffer cache size, hence it divides buffer cache into two equal parts. The first part of the buffer cache starting with HOT END is called as HOT REGION and the remaining part ending with COLD END is called as COLD REGION. The other END of the HOT REGION opposite to HOT END is called as WARM END.
- In order to keep adaptability with changing reference patterns, we maintain a LRU list (LIST1) of recently replaced pages from the buffer cache. The length of the list is same as the length of the buffer cache.

5.4 Accurate Probabilistic Adaptive Clean First Algorithm (APR-ACF)

When page b_i is referred at time T_i then following procedure is called (assume buffer full condition). We call this algorithm as Accurate Probabilistic Adaptive Clean First Algorithm (APR-ACF).

Replacement (b_i,T_i)
Begin
If (b_i is not in buffer cache)

1. scan the COLD REGION from the COLD END towards the WARM END for the first CLEAN page.
   - if found then replace first CLEAN page with b_i. (a)
   - if not found then replace the COLD END page with b_i.
2. Add replaced page to MRU end of LIST1. If LIST1 is already FULL, delete the LRU END page from it, and add replaced page to MRU end. (a1)
3. Update probabilities of each page with new time reference T_i. (b)
4. Sort the list. (c)

5. resize() (d)
6. **End**

Following procedure resize() adapts with the changes in the reference patterns of the working set pages and normal pages. This is achieved by resizing the COLD and HOT regions at each page reference as follows:

resize ()

- If referred page b_i is a working set page and is present in the LIST1 (list of recently replaced pages), then update variable SPLIT to decrease length of COLD REGION by 5 % of the buffer cache length. This will increase the length of HOT REGION by the same 5 %.
- If referred page b_i is a normal page (does not belong to working set of current time interval) and is present in the LIST1, then update variable SPLIT to increase length of COLD REGION by 5 % of the buffer cache length. This will decrease the length of HOT REGION by the same 5 %.
- The range of values that SPLIT variable can take are 10 %–90 % of the buffer cache length (boundary condition). In case both the above two conditions fails or increase/decrease of lengths of buffer cache parts causes violation of boundary condition, the COLD and HOT regions will not be resized.

Step (a) executes in $O(1)$ time. Step (b) is executed in $O(k)$ time while Step (c) is executed in $O(k \log k)$ time where k is size of buffer cache. Step (d) is an adaptive step which responds quickly to changes in page reference patterns (can take place in constant time). Adaptation occurs whenever the page fault corresponds to a page which is recently replaced. In case it is a working set page, the length of HOT region will be increased. In other case when it is a normal page, length of COLD region will be increased. Our algorithm assumes that in the cache most of the working set pages will be present in the HOT region and most of the normal pages will be present in the COLD region. Generally, working set pages will join the buffer cache in the HOT region, whereas normal pages will join the buffer cache in the COLD region. Afterwards, based on their relative reference patterns with respect to other cached pages, they will move towards the HOT end or COLD end of the buffer cache. Step (a1) can also happen in a constant time. The proposed algorithm is capable of providing performance for most of the common referencing patterns which is explained as follows.

Sequential References. In a sequential scan, pages are referenced and processed one after another. For non repeated scan, the probability of buffer goes on decreasing and finally buffer is edged out of the cache. In case of repeated scan the frequency of reference is increased and according to Equation (9) the probability is increased which will increase stay of page in the buffer cache. In the case of clustered sequential access (CS) like merge join, the buffer is frequently referred and according to Equation (9) its probability is increased.

Hierarchical References. This reference behavior is observed where index is repeatedly used, non leaf nodes of the index tree are referred frequently and their probability of stay is increased according to the Equation (9).

Infrequent Long Query. If the query is infrequent and requires lot of disk page access then to bring the required disk pages in the cache, whole existing buffers will be replaced. This will lead to removal of existing and stable working set in the cache and can result in lot of cache misses afterwards. The pages of such long infrequent queries will not appear in the working set due to their infrequent access pattern and they are inserted towards COLD END and subsequently they will be replaced quickly without disturbing the stable working set.

6 COSTS BENEFIT ANALYSIS

The above proposed algorithm (APR-ACF) enhances flash I/O performance by increasing the hit ratio and minimizing the write-erase operations, but the cost of replacement policy is very high because it involves updating probabilities and sorting list. The overall performance of the system can be analyzed by considering overheads in replacement strategy and improved hit ratio as explained in following section.

Suppose

- C: size of buffer cache,
- B: hit ratio,
- N: number of memory references made by replacement algorithm,
- M: time required for one memory access,
- D: time required for one disk access,
- B_0: minimum hit ratio assuming elementary replacement policy,
- B_1: maximum hit ratio gain by most complex algorithm.

Normally hit ratio is improved if more information is used to decide replacement buffer, but the improvement is not linear. Practically after reaching certain value there will be only marginal increase in hit ratio even if large information is scanned for deciding replacement buffer. Thus the gain in hit ratio gain is inversely proportional to the hit ratio hence the relationship can be approximated using following equation:

$$B = B_0 + B_1(1 - \exp(-\alpha_i nC)) \tag{10}$$

where $0 < B_0 + B_1 < 1$.

Assuming replacement decision is made by analyzing n bytes of information and it is repeated C times, i.e. for each buffer in the cache. The α_1 is proportionally constant which is approximately equal to improvement in hit ratio by scanning additional one information byte. Here exponential function is used because the hit ratio will increase less as proportion to increase in complexity of the replacement algorithm, and after certain value it cannot be increased by the most complex replacement algorithm.

The cost of replacement policy depends on number of information bytes scanned in deciding replacement. This cost is linearly proportional to number of information bytes scanned.

Cost of replacement strategy $= (\alpha_0 + \alpha_2 nC)$ where α_0 is minimum bytes scanned for each reference and α_2 is proportionality constant.

If there are N references, then the total access time (T_0) with elementary replacement strategy is given by following equation:

$$T = N\left(M + (1 - B_0)D\right) + N\alpha_0 M = N\left(M + (1 - B_0)\gamma M\right) + N\alpha_0 M \tag{11}$$

where $D = \gamma M$.

According to current technology parameters, $\gamma \gg 10^6$.

When replacement is done using complex algorithm by scanning n bytes of information then the total access time (T) is

$$\begin{aligned} T_0 &= N\left(M + (1 - B)\gamma M\right) + NM\left(\alpha_0 + \alpha_2 nC\right) \\ &= N\left(M + (1 - B_0 - B_1 \exp(-\alpha_i nC))\gamma M\right) + MN\alpha_0 + N\alpha_2 nCM, \end{aligned} \tag{12}$$

$$T = T_0 - N\gamma B_1 \exp(-\alpha_i nC)M + N\alpha_2 nCM, \tag{13}$$

$$T = T_0 - T_- + T_+ \tag{14}$$

where

- T_- is reduction in time due to improved hit ratio due to complex replacement strategy,
- T_+ is increase in replacement time due to complex replacement strategy.

The overall reduction in time requires, $T_- \gg T_+$

$$\gamma B_1 \exp(-\alpha_1 nC) \gg \alpha_2 nC. \tag{15}$$

If γ is more than n, then the proposed algorithm always gives good performance, however, if buffer cache size is very large then n is also higher because information of more buffers is to be kept and cost of replacement policy tends to be very high making proposed algorithm less practically feasible. The cost can be reduced by avoiding updating and sorting list for each reference.

By using efficient data structures n can be reduced and inequality given in Equation (15) can be satisfied. To reduce cost of replacement we are proposing approximate algorithm which is having reduced replacement cost without much decreasing hit ratio.

7 APPROXIMATE ALGORITHM

To avoid modification of probabilities in step (b) and sorting of the list in step (c) of the above mentioned proposed algorithm, only historical count (HC) in the interval time of the current reference and frequency count (FC), which is the number of references in current time in that interval, are maintained. The total count (TC) is calculated as sum of historical count and frequency count which is not changed unless page is referred so the list will remain sorted and page which is referred is always inserted to maintain it in sorted order. The modified algorithm is:

Replacement (b_i, T_i)
If (b_i is not in buffer cache)
$b_i.FC = 0$;
$b_i.HC =$ (count from historical list if it exists in historical list, otherwise 0);

- scan the COLD REGION from the COLD END towards the WARM END for the first CLEAN page.
  - if found then replace first CLEAN page with b_i. (a2)
  - if not found then replace the COLD END page with b_i.
- Add replaced page to MRU end of LIST1. If LIST1 is already FULL, then delete the LRU END page from it, and add replaced page to MRU end. (a22)

$b_i.FC = b_i.FC + 1$; (b21)
$b_i.TC = b_i.TC + 1$; (b22)
$cnt++$;
Insert b_i in page_replacement_list in sort order; (c2)
If ($cnt >$ threshold) decrease FC and HC of each page; (c3)
resize() (d)
End

Steps a2, b21, b22 are executed in $O(1)$ times and step (c2) is executed in $O(\log k)$ times where k is size of buffer cache. If there is page in page_reference_list having the same total count then b_i is always inserted towards hot end in the sorted order.

If the reference pattern shows periodicity of references then historical count is increased and page is moved towards hot end. If the page is referred frequently then its current count is increased and it is moved towards hot end. If the page is referred by infrequent query then it is inserted towards cold end and it is finally moved out of buffer cache. If the page is referred frequently initially but its recent references are very less then it is moved towards cold end because of step (c3). In the COLD REGION likelihood of replacement will be high. The cost of replacement policy depends on step (c3) and can be reduced by increasing threshold. The algorithm finally guarantees longer stay to non-cold DIRTY pages and HOT CLEAN pages and a shortest possible stay to COLD CLEAN pages, followed by warm CLEAN pages and COLD dirty pages.

8 EXPERIMENTATIONS

For performance evaluation, we compare the best of our two algorithms, i.e., PR-ACF with the performance of five best competitors, namely CF-LRU, LRU-WSR, CCF-LRU, AD-LRU and PR-LRU. The performance measures used are buffer hit ratio, number of write operations and runtime.

8.1 Experimental Environment

The simulation experiments are conducted based on flash memory simulation framework, called FlashDBSim. FlashDBSim is a reusable and reconfigurable framework for the simulation-based evaluation of algorithms on flash disks [23]. FlashDBSim uses a modular design approach, which includes Virtual Flash Device Module (VFD), Memory Technology Device Module (MTD), and Flash Translation Layer Module (FTL). The VFD module is a software layer that simulates the actual flash memory devices. Its most important function module is to provide virtual flash memory using DRAM or even magnetic disks. It also provides manipulating operations over the virtual flash memory, such as page reads, page writes, and block erases. The MTD module maintains a list of different virtual flash devices, which enables us to easily manipulate different types of flash devices, e.g., NAND, NOR, or even hybrid-flash disks. The FTL module simulates the virtual flash memory as a block device so that the upper-layer applications can access the virtual flash memory via block-level interfaces. The FTL module employs the EE-Greedy algorithm citeref14 in the garbage collection part and uses the threshold for wear-leveling proposed in [6]. In our experiment, we simulate a 128 MB NAND flash device with 64 pages per block and 2 KB per page. The I/O characteristics of the flash device are shown in Table 1 and the erasure limitation of blocks is 100 000 cycles.

Operation	**Access Time**	**Access Granularity**
Read	20 μs/page	Page (2 KB)
Write	200 μs/page	Page (2 KB)
Erase	1.5 ms/block	Block (128 KB = 64 pages)

Table 1. The characteristics of NAND flash memory

8.2 Dataset Characteristics

We have performed a trace-based simulation to evaluate the performance of the proposed PR-ACF algorithm in comparison with the competitor algorithms. We have done the experimentation on four different traces of 24×7 days which contain a mixture of random, sequential and repetitive patterns along-with different read and write localities. The first six days traces are used as a training dataset to define working sets for different time intervals and seventh days trace is used as a test dataset for performance evaluation. The details of the traces are given in Tables 2 and 3.

Trace-ID	No. of Refs	No. of Pages	Read/Write Ratio	Locality
T1	250 000	12 000	80 %/20 %	60 %/40 %
T2	250 000	12 000	20 %/80 %	40 %/60 %
T3	250 000	12 000	50 %/50 %	80 %/20 %
T4	250 000	12 000	60 %/40 %	80 %/20 %

Table 2. Characteristics of four traces Set-A

A read/write ratio "X %/Y %" in Table 2 means that the read and write operations in the traces are of X and Y percentages, respectively. The locality expression in Table 2, e.g. X %/Y %, means that X % of total number of accesses call Y % of total number of data pages. Hence the likelihood of our working sets having members as the subset of these Y % data pages is very high.

Another workload that we have used for experimentation is OLTP one hour test trace in a real bank system containing 607 391-page references to a CODASYL database with a total size of 20 Gigabytes. The number of different pages accessed is 51 870 with each page having the size of 2 048 bytes. Ratio of read/write operations is 77 %/23 %. Table 3 gives the distribution of various types of references in each workload trace.

8.3 Performance Metrics

Three performance metrics, write count, hit ratio, and runtime were used in our simulation experiments to evaluate the results. The erase operations are not considered because the erase counts are nearly proportional to the write counts, as they are triggered due to call to write operations.

The read operations are not considered because firstly reads are covered in the hit ratio parameter and secondly they are less significant to overall performance due to its low latency compared to write operations.

Runtime parameter is highly influenced by hit ratio and the number of writes to the flash memory.

Trace-ID	Periodic Refs	Sequential Refs	Repetitive Refs	Random Refs
T1	7.22 %	14.43 %	60.26 %	18.09 %
T2	10.47 %	18.33 %	24.51 %	46.69 %
T3	19.32 %	21.62 %	29.58 %	29.48 %
T4	3.47 %	57.81 %	21.42 %	17.30 %

Table 3. Characteristics of four traces Set-B

8.4 Parameter Settings

For all the datasets, parameter w of the CFLRU algorithm is set to 0.5, which means half of the buffer is used as clean-first window. Parameter min_lc of AD-LRU is set to 0.2.

8.5 Results and Results Analysis

Figures 1–4 illustrate the comparison of the hit ratios on traces T1 to T4 for various buffer sizes.

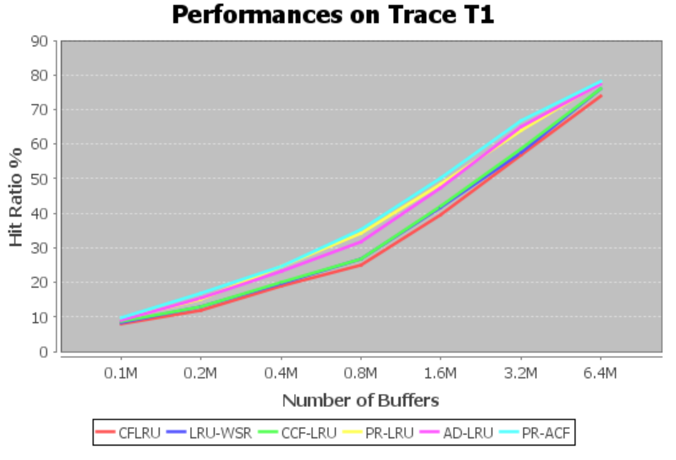


Figure 1. Hit ratio comparison on trace T1

On all the four traces, PR-ACF has a better hit ratio than the other algorithms, as shown in Figures 1–4. PR-ACF considers the locality of pages from recent as well as historical references. At each replacement, it selects the page from the COLD region, and within the COLD region, it prefers the COLDEST clean page for a replacement. In absence of CLEAN page in the COLD region, it replaces the COLDEST dirty page. Here, the COLD page corresponds to a page having a low probability of reference in the near future. Hence using this COLD first policy PR-ACF has achieved the best hit ratio in read-most as well as write-most scenarios, as the replaced page has a high probability of not getting referred shortly. On traces with a high percentage of random page references, PR-ACF manages to outperform all the competitor algorithms, because of its online adaptivity in which it continually controls the growth in the miss rate by resizing HOT and COLD regions, according to changes in the page access patterns. One of the important advantages of probabilistic cache is that pages in sequential reads and sequential writes are directly inserted in the COLD region hence quickly replaced, eliminating the possibility of cache pollution. Pages in random

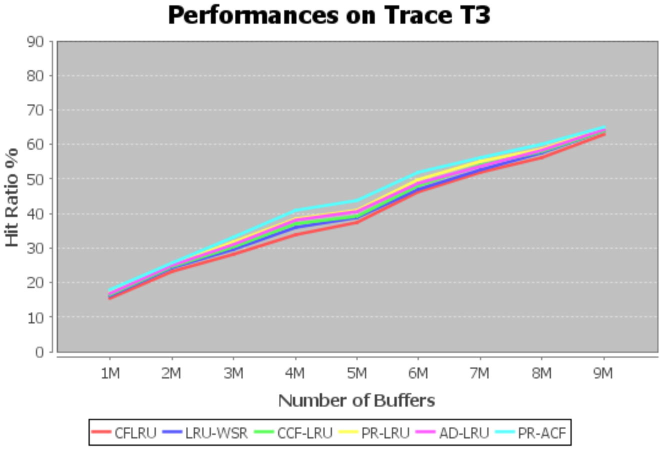


Figure 2. Hit ratio comparison on trace T2

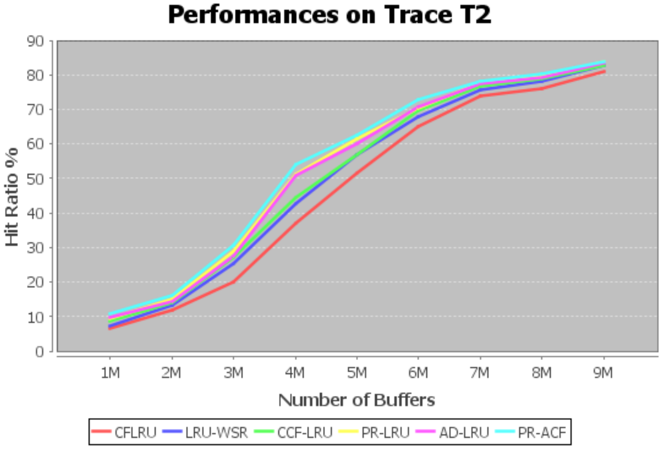


Figure 3. Hit ratio comparison on trace T3

reads and random writes also will have a short stay in the buffer cache, whereas pages in periodic and repetitive reads-writes will have longer stay in the buffer cache.

The other way around, many of the competitor algorithms evicts clean pages without considering their access frequencies, to protect dirty pages. Few of them protect HOT clean pages by selecting COLD dirty pages for replacement, but they consider only recent access patterns to predict the HOT and COLD pages. This adversely affects their hit ratio, what is the reason why our proposed algorithm outperforms them considerably. The increase in the hit ratio by PR-ACF when

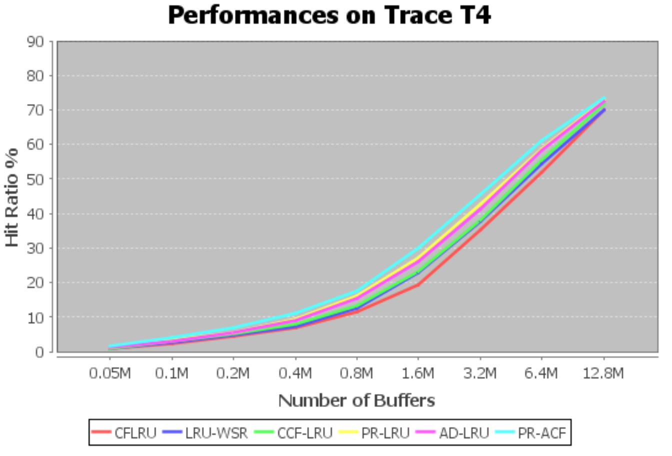


Figure 4. Hit ratio comparison on trace T4

compared with other best-performing algorithms CCF-LRU, AD-LRU and PR-LRU is 6.3 %, 4.5 %, and 2.7 %, respectively, on trace T3.

Figures 5–8 show the number of write operations for each algorithm on traces T1 to T4 for different buffer sizes. As shown in Figures 5–8, the write count of PR-ACF is less than all the competitor algorithms on all the traces. The primary reason for this is that all the HOT dirty pages (having close to largest write and erase cost) are saved from getting replaced, as all the HOT (having high probability of access at current instance of time) pages are in the HOT region, a region which is forbidden for replacement in PR-ACF. In the COLD region, PR-ACF favors evicting clean pages first from the buffer so that the number of writes incurring from replacements of COLD dirty pages can be reduced.

In absence of clean page in the COLD region, PR-ACF replaces the COLDEST dirty page, thereby protecting the HOT clean pages from replacement, and avoiding unnecessary degradation of the hit ratio. As COLDEST dirty page will be having the lowest probability of reference in the cache, there is less chance of it getting referred and causing an increase in the number of write counts in the current time interval. As the time interval changes so as the working set and the decisions of PR-ACF about which pages to protect from replacement also changes. Hence the above working principle benefits PR-ACF in all the three read-most, write-most and random-most scenarios in maintaining the low write count and high hit ratio simultaneously. The reduction in the number of writes by PR-ACF when compared with other best-performing algorithms CCF-LRU, AD-LRU and PR-LRU is 42.7 %, 29.6 %, and 24.2 %, respectively, on trace T2.

Figures 9–12 show the overall runtime of various replacement algorithms. The runtime of an algorithm is the sum of time required for read, write and erase operations plus the memory time. The access time for each type of operation is given

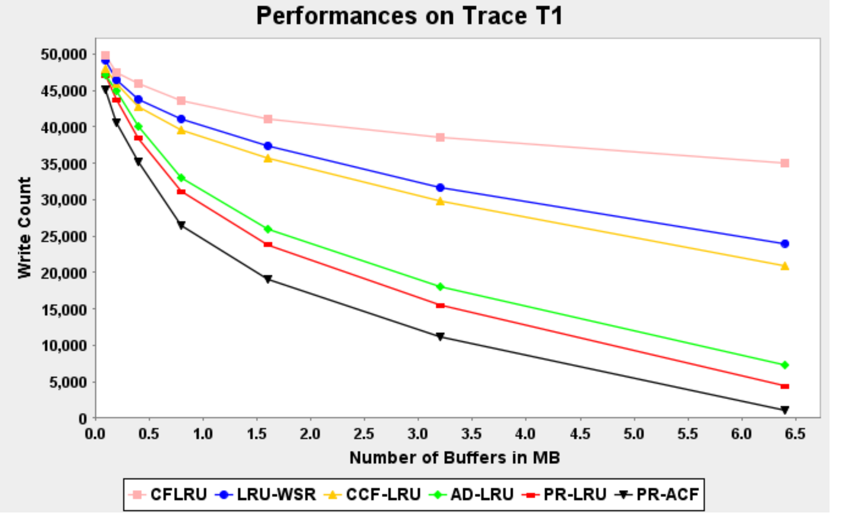


Figure 5. Write count comparison on trace T1

in Table 1. The total runtime of an algorithm can also be calculated as a number of read operations ∗ read access time + a number of write operation ∗ write access time + a number of erase operation ∗ erase access time. The total runtime of an algorithm is highly influenced by its hit ratio and the number of write operations (write count) involved.

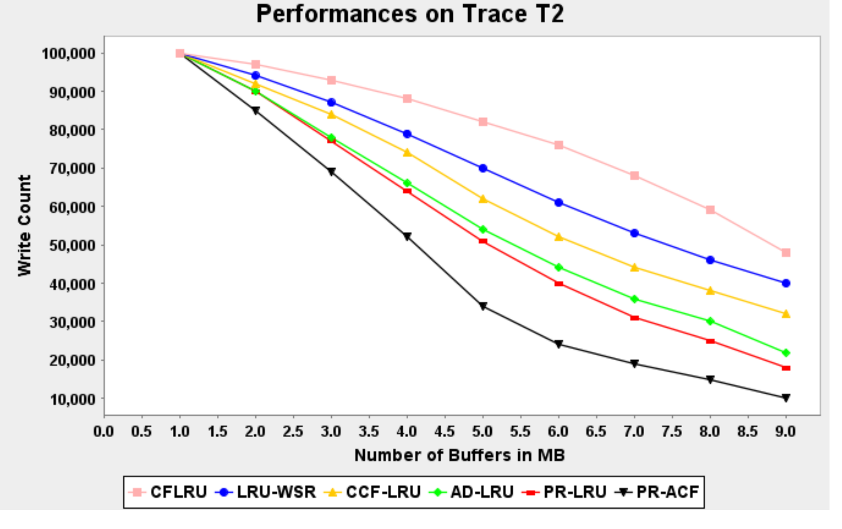


Figure 6. Write count comparison on trace T2

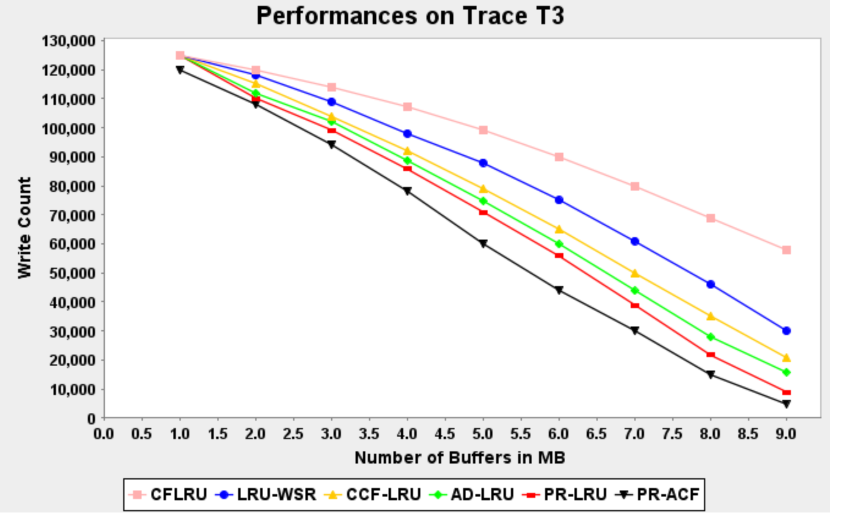


Figure 7. Write count comparison on trace T3

Specifically, total runtime is directly proportional to the write count and inversely proportional to the hit ratio. Figures 9–12 show that PR-ACF has the lowest runtime. This is because PR-ACF maintains the highest hit ratio and lowest write count amongst all the algorithms. The reduction in the runtime by PR-ACF when compared with other best-performing algorithms CCF-LRU, AD-LRU and PR-LRU is 30.4 %, 7 %, and 5.1 %, respectively, on trace T4.

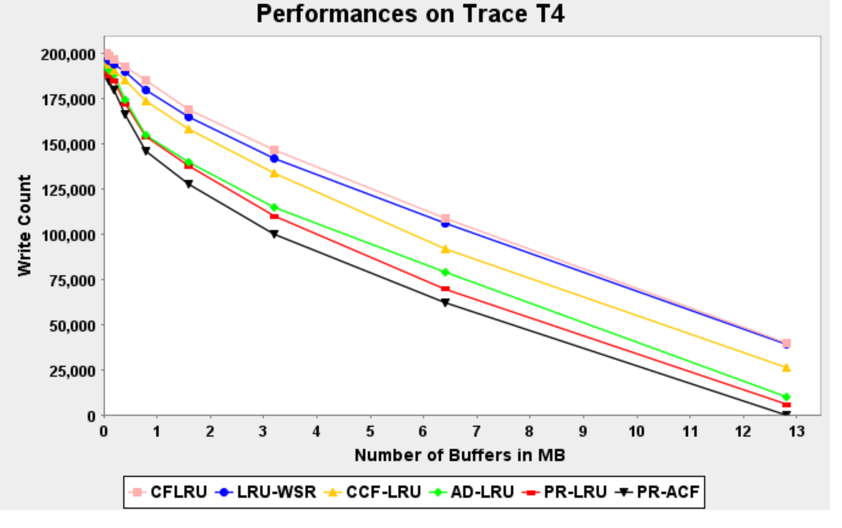


Figure 8. Write count comparison on trace T4

In most of the cases, total runtime is less influenced by running time of the algorithm (also called memory time) as compared to the I/O operational time (time taken by read, write and erase operations). However, with the weak locality, memory time has a greater impact on the runtime. With the increasing buffer cache size, the ratio of increase in memory time and decrease in total runtime keeps on increasing.

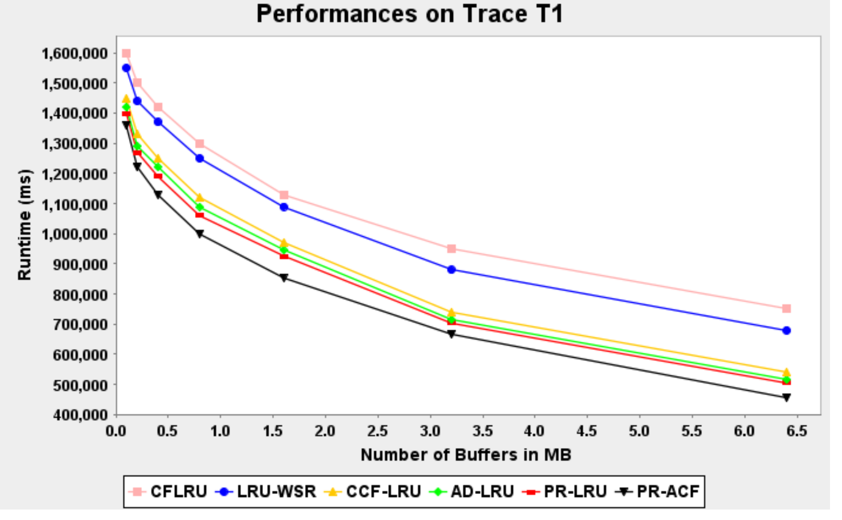


Figure 9. Runtime comparison on trace T1

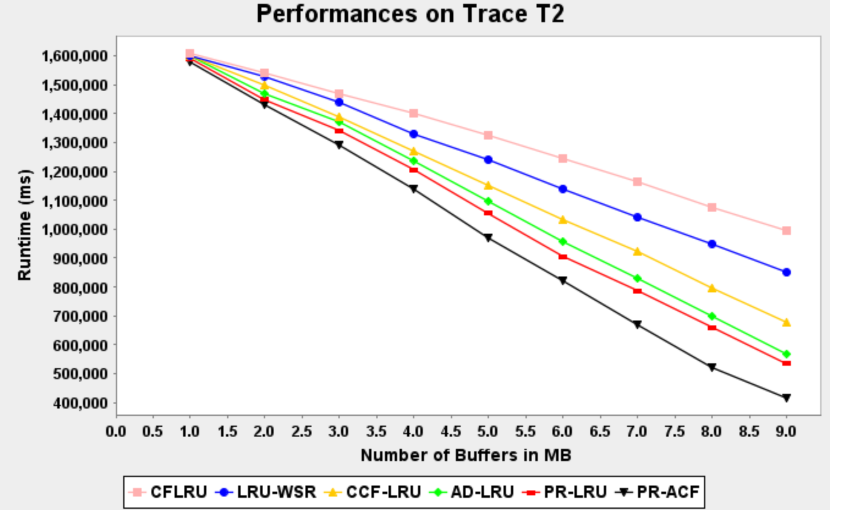


Figure 10. Runtime comparison on trace T2

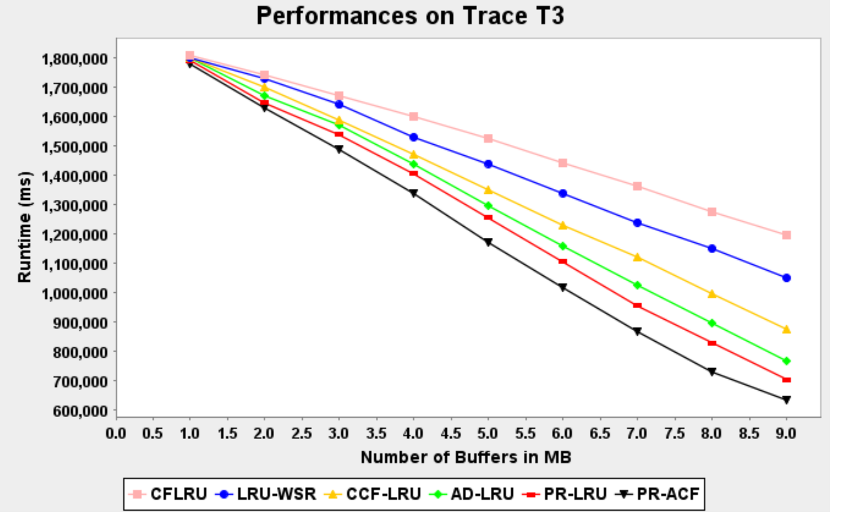


Figure 11. Runtime comparison on trace T3

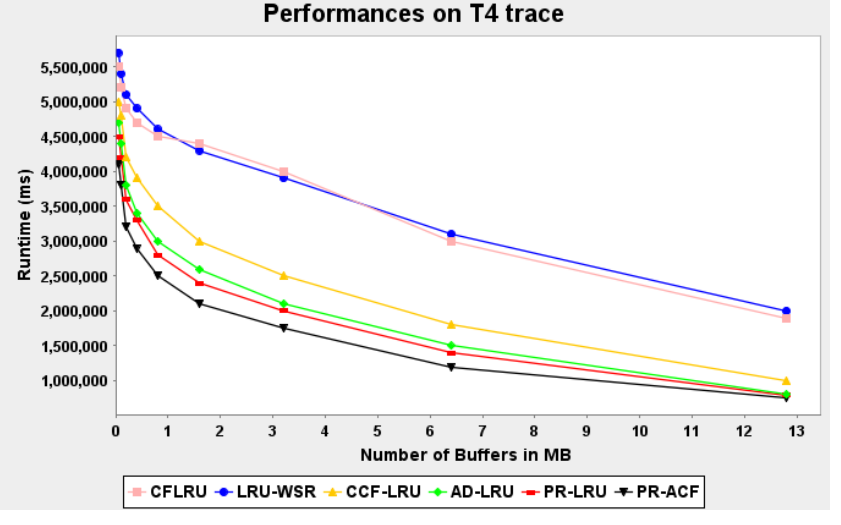


Figure 12. Runtime comparison on trace T4

Figures 13, 14, 15 show the hit ratio, write count and runtime comparison, respectively, of various replacement algorithms on real OTLP trace. PR-ACF has a clear advantage over other algorithms in terms of hit ratio, write count and runtime for most of the buffer cache sizes.

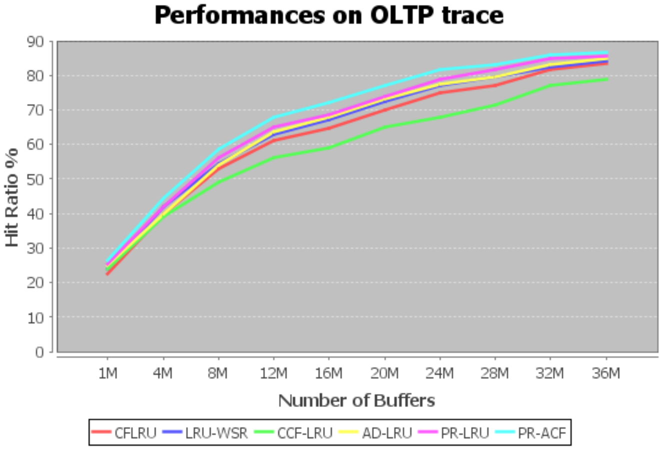


Figure 13. Hit ratio comparison on OLTP trace

9 OPTIMAL SET OF TIME INTERVALS

The above-proposed method uses static intervals like 9–10 am, 10–11 am and so on for calculating the predictive working set. This will give a simple and less computational algorithm for replacement but it may not give optimal predictive working set. Calculating correct intervals to get an optimized predictive working set and to get

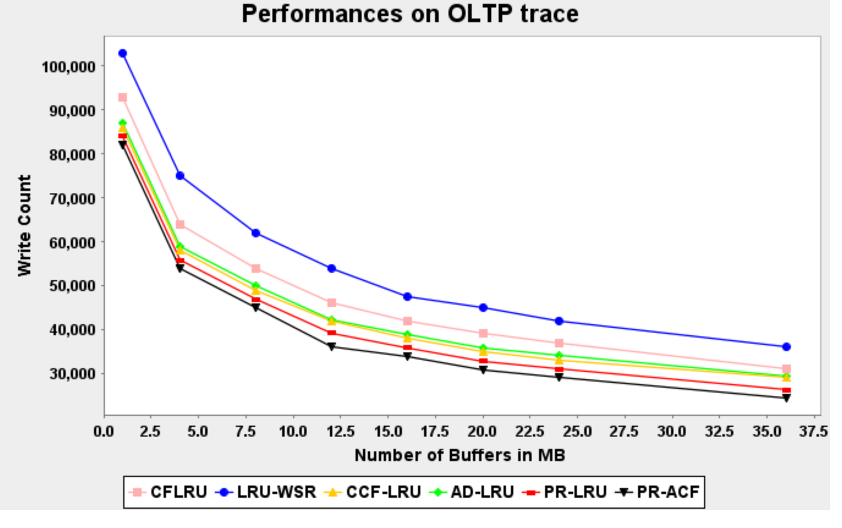


Figure 14. Write count comparison on OLTP trace

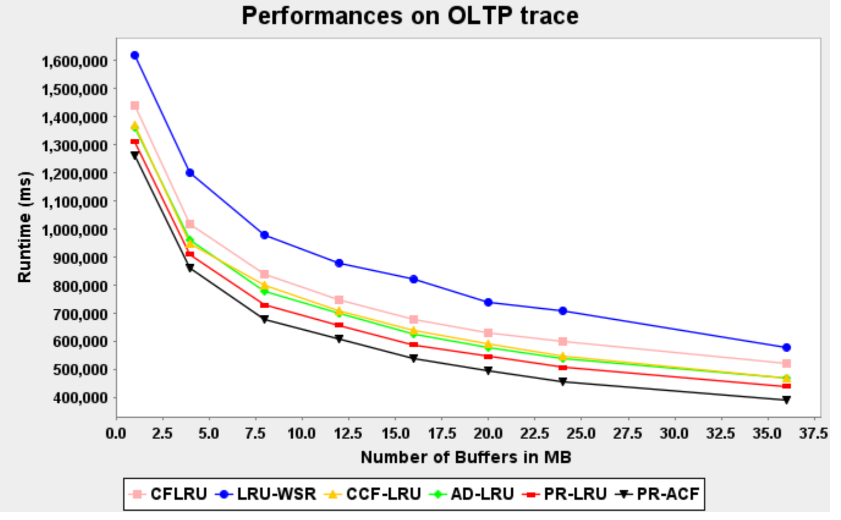


Figure 15. Runtime comparison on OLTP trace

minimum cache misses is a challenging task. Since it is a combinatorial optimization problem, techniques like genetic algorithms can be used.

9.1 Genetic Algorithms

Genetic Algorithms (GAs) are adaptive heuristic search algorithms based on the evolutionary ideas of natural selection and genetics. As such, they represent an intelligent exploitation of a random search used to solve optimization problems. After an initial population is randomly generated, the algorithm evolves through three operators:

- Selection, which equates to the survival of the fittest.
- Crossover, which represents reproduction by the crossover between solutions.
- Mutation, which introduces random modifications.

9.2 Proposed Genetic Algorithm for Finding Optimal Set of Time Intervals

For finding an optimal set of time intervals we have defined a system that can generate a solution under specified constraints. Initially, it reads values for the following three input parameters.

- Total number of time intervals in a solution (N).
- The minimum length of the time interval (minlength).

Algorithm 1 Generic Genetic Algorithm

```
randomly initialize population (p)
determine fitness of population (p)
while best individual is not good enough or number of evolutions does not reach
its limiting value do
    select parents from population (p)
    perform crossover on parents creating population (p + 1)
    perform mutation of population (p + 1)
    determine fitness of population (p + 1)
end while
```

- The maximum length of the time interval (maxlength).

After receiving these 3 inputs, the system randomly creates an initial population of M solutions (P) in which each solution has exactly N time intervals and the length of each time interval is between minlength and maxlength. Each solution needs to cover a complete daytime of 24 hours represented by 1 to 1 440 (1 min to 1 440 min $(24 * 60)$). Where 1 represents 00:01 and 1 440 represents 00:00.

Example 1. Each solution of the sample solution set has the following characteristics: $N = 12$, minlength $= 80$, maxlength $= 160$. In the initial population of 100 solutions ($M = 100$) one of the solution is given in Table 4.

Interval-ID	Start-Time	End-Time	Fitness
1	1 065	1 148	0.56
2	1 148	1 283	0.64
3	1 283	1 419	0.64
4	1 419	66	0.58
5	66	178	0.66
6	178	305	0.6
7	305	415	0.72
8	415	538	0.68
9	538	672	0.63
10	672	826	0.7
11	826	959	0.66
12	959	1065	0.71
Solution: 21	Total Fitness		7.78

Table 4. Sample solution

The proposed algorithm works as follows: Within the minimum and maximum length constraints on the duration of the time interval, our algorithm randomly generates a population of M solutions each having N time intervals. M and N are predefined parameters that remain constant throughout the evolutionary process. Fitness of each solution is computed by taking the summation of the fitness of its all the N time intervals. Fitness of each time interval is computed using the computeFitness function which takes the starttime and endtime of the time interval as input parameters and returns the summation of access probabilities of the top 15 % pages (working set pages) in that time interval. The information about pagewise reference probabilities for a given time interval is derived from the historical data by applying the probabilistic model discussed in Section 4.

An initial population P of M solutions is generated through the 3 steps of Algorithm 2.

9.2.1 Generating Initial Population of 100 Solutions

9.2.2 Finding Fairly Optimal Solution

The more the fitness of the solution, it can be expected that the higher is the overall daywise benefits to the cache performance from the N working sets of the solution. Here the N working sets of a solution are associated with N different intervals of time (of unequal duration) having different fitness values (For example refer to Table 4). To maximize the overall daywise benefits to the cache performance, i.e., to maximize the cache optimization, we search heuristically for the close to an optimal solution. For that the initial population of M solutions evolves through T iterations, using the selection, mutation and crossover operators (described in Section 9.3). After T iterations, in the final population of solutions, the solution with the highest fitness value is to be selected as close to an optimal solution (fairly optimal solution).

9.3 Genetic Operators

9.3.1 Selection

This operator picks the top 25 % solutions (in terms of fitness) from the initial/previous population for reproduction. The top 25 % solutions (in terms of fitness) from initial or previous population P are copied into the new population P'. The rest of the solutions in the new population P' are generated by crossover (50 % solutions) and mutation (25 % solutions) operators.

9.3.2 Crossover

This operator picks randomly any 2 solutions parent1 and parent2 from P and creates a new child solution having combined features of parent1 and parent2. The child or a new solution is included in the new population P'. We compute Avg1 and Avg2 as the average fitness value of parent1 and parent2 respectively.

$$\text{Avg1} = \text{Total Fitness of parent1}/N,$$

$$\text{Avg2} = \text{Total Fitness of parent2}/N.$$

The new – child solution will also have total N time intervals. The combined features (best of each of them) of parent1 and parent2 are copied into a new solution in the following way: The endtime of i^{th} time interval of parent1 is copied into the endtime of i^{th} time interval of a new solution if parent1 has the best solution for the interval I, i.e., interval I of parent1 satisfies the following conditions:

- Its fitness value > than Avg1 as well as Avg2.
- Its fitness value > fitness value of i^{th} time interval of parent2.

The starttime of i^{th} time interval of new solution is set to endtime of the previous – $(i-1)^{\text{th}}$ interval of the new solution or to the starttime of i^{th} time interval of parent1 if $i = 1$ (first interval).

A similar policy is employed for copying best fitness intervals of parent2 into a new solution. For the time interval I if neither parent1 nor parent2 has the best solution, we follow the following strategy: The starttime of i^{th} time interval of new solution is set to endtime of the previous – $(i-1)^{\text{th}}$ interval of the new solution or to the average of starttime of i^{th} time interval of parent1 and parent2 (for the first interval of a new solution).

The endtime of i^{th} time interval of new solution is set to the average of endtime of i^{th} time interval of parent1 and parent2.

In case of violation of interval length constraint in computing endtime in the above way, the endtime is recomputed using formula (16).

$$\text{endtime} = (\text{starttime} + \text{random}(\text{maxlength} - \text{minlength}) + \text{minlength})\,\%\text{high}. \quad (16)$$

9.3.3 Mutation

This operator selects the top 25 % solutions in terms of fitness from population P and adds them to empty set S. Each solution from set S is mutated randomly before adding it to a new population P'. The mutation is carried out in the following way.

From the solution find the top $N/4$ time intervals with the lowest fitness value. Mutate these intervals, i.e., regenerate the endtime of each of these intervals using formula (16). Recompute their fitness and add them to the mutated solution. If mutated intervals cause violation of interval length constraints for the subsequent intervals regenerate the endtimes of subsequent intervals using formula (16), up to the interval for which violation stops.

9.3.4 Experimental Results

We have taken a seven days OLTP traces from the commercial MIS database server. These traces are timestamped and have the page size of 2 KB. After analyzing, the set of intervals reported by the proposed genetic algorithm (Number of Iterations = 100, Population Size = 100, Number of Intervals = 11) is as follows:

31–158–303–425–520–695–829–937–1 099–1 254–1 382–31.

Algorithm 2 Generating initial population of 100 solutions

Step 1: ▷ Parameters which are constant for all the solutions in the population are initialized.
$M = 100$ ▷ No of solutions in the initial population
Low = 1 ▷ represents 00:01 hrs
High = 1 440 ▷ represents 00:00 hrs
Read N ▷ number of time intervals in each solution
Read minlength ▷ minimum length of each time interval
Read maxlength ▷ maximum length of each time interval
[end of Step 1]
Step 2: ▷ create initial population P
for $I = 1$ to M repeat Step 3.
[end of Step 2]
Step 3: ▷ create I^{th} solution in P
Solution (I).Interval (1).Starttime = Low + random(High − Low); ▷ Set starttime of first interval randomly by value between the range Low to High.
Solution (I).Interval (1).Endtime = (Solution (I).Interval (1).Starttime + random (maxlength − minlength) + minlength) % High; ▷ Set the endtime of first interval by adding a random value between minlength and maxlength to the starttime of first interval.
Solution (I).Interval (1).Fitness = computeFitness (Solution (I).Interval (1).Starttime, Solution (I).Interval (1).Endtime); ▷ Compute Fitness of first time interval of Solution I
for $J = 2$ to N repeat Step 4. ▷ Repeat the process for remaining $N - 1$ time intervals of Solution I, all the time intervals will have starttime equal to endtime of the previous time interval.
[end of Step 3]
Step 4:
Solution (I).Interval (J).Starttime = Solution (I).Interval ($J - 1$).Endtime;
Solution (I).Interval (J).Endtime = (Solution (I).Interval (J).Starttime + random (maxlength − minlength) + minlength) % High;
Solution (I).Interval (J).Fitness = computeFitness (Solution (I).Interval (J).Starttime, Solution (I).Interval (J).Endtime); ▷ Compute Fitness of J^{th} time interval of Solution I
[end of Step 4]
Step : ▷ Compute Fitness of a solution Solution(I).Fitness = $\sum_{j=1}^{N}$ [Solution(I).Interval(J).Fitness]
[end of Step 5]

We defined working sets for the above optimal set of time intervals as well as working sets for the static time intervals of one-hour fixed length on the above dataset.

Here a new working set (corresponding to new time interval) is loaded automatically at the expiration of each time interval (which coincides with the start of the next time interval).

Experimental results prove that the use of optimal time intervals in PR-ACF maximizes the overall day-wise performance gains considerably in comparison to the results achieved with static time intervals (one hour each), as shown in Figure 16.

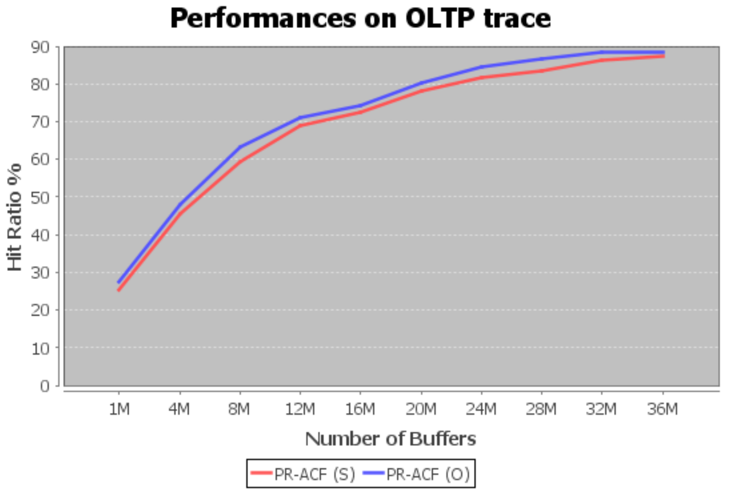


Figure 16. Performance comparison of PR-ACF algorithm with static (S) and optimal (O) time intervals

10 CONCLUSION

In this paper, we focus on a cache replacement policy for database systems equipped with flash memory as a secondary storage. We propose a new replacement policy, called PR-ACF, which considers the imbalance of read and write costs of flash memory while replacing pages. The basic idea behind PR-ACF is to avoid the replacement of dirty pages present in the buffer cache to minimize the number of write operations and at the same time preventing the significant degradation in the hit ratio to achieve the fairly close to overall optimal performance. To determine the coldness and hotness of the cached pages we propose a probabilistic model that calculates the probability of future reference of each cached page and organizes the cache based on computed probability. To improve the accuracy of prediction, the probability is calculated based on the study of reference patterns from a history of references along with recent reference patterns. The buffer cache is divided into

HOT and COLD regions which are dynamically resized according to the changing access patterns. The page replacement always happens in the COLD region, and within the COLD pages, the COLDEST clean page is targeted for replacement, thus deliberately keeping the COLD dirty pages in the cache to avoid performance degradation due to costly write and erase operations on flash memory. The proposed PR-ACF algorithm was tested on the flash simulation platform Flash-DBSim, by comparing its performance with best-known flash-based replacement algorithms. The experimental results show that our proposed algorithm performs better than the top-performing algorithms like CCF-LRU, AD-LRU, and PR-LRU with respect to write count, runtime as well as hit ratio. Experimental results also prove that the use of optimal time intervals maximizes the overall day-wise performance gains considerably in comparison to the results achieved with static time intervals.

REFERENCES

[1] Robinson, J. T.—Devarakonda, M. V.: Data Cache Management Using Frequency-Based Replacement. ACM SIGMETRICS Performance Evaluation Review, Vol. 18, 1990, No. 1, pp. 134–142, doi: 10.1145/98460.98523.

[2] Chou, H.-T.—DeWitt, D. J.: An Evaluation of Buffer Management Strategies for Relational Database Systems. Algorithmica, Vol. 1, 1986, No. 1-4, pp. 311–336, doi: 10.1007/BF01840450.

[3] Sacco, G. M.—Schkolnick, M.: Buffer Management in Relational Database Systems. ACM Transactions on Database Systems, Vol. 11, 1986, No. 4, pp. 473–498, doi: 10.1145/7239.7336.

[4] Pan, Z.-W.—Xiang, D.-H.—Xiao, Q.-W.—Zhou, D.-X.: Parzen Windows for Multi-Class Classification. Journal of Complexity, Vol. 24, 2008, No. 5-6, pp. 606–618, doi: 10.1016/j.jco.2008.07.001.

[5] Shen, H.—Yan, X.-L.: Probability Density Estimation over Evolving Data Streams Using Tilted Parzen Window. 2008 IEEE Symposium on Computers and Communications, 2008, pp. 585–589, doi: 10.1109/ISCC.2008.4625751.

[6] Lofgren, K. M. J.—Norman, R. D.—Thelin, G. B.—Gupta, A.: Wear Leveling Techniques for Flash EEPROM Systems. United States Patent US6850443B2, 2005.

[7] Park, S.-Y.—Jung, D.—Kang, J.—Kim, J.—Lee, J.: CFLRU: A Replacement Algorithm for Flash Memory. Proceedings of the 2006 International Conference on Compilers, Architecture and Synthesis for Embedded Systems (CASES '06), ACM, 2006, pp. 234–241, doi: 10.1145/1176760.1176789.

[8] Jin, P.—Härder, T.—Li, Z.: AD-LRU: An Efficient Buffer Replacement Algorithm for Flash-Based Databases. Data and Knowledge Engineering, Vol. 72, 2012, pp. 83–102, doi: 10.1016/j.datak.2011.09.007.

[9] Yuan, Y.—Shen, Y.—Li, W.—Yu, D.—Yan, L.—Wang, Y.: PR-LRU: A Novel Buffer Replacement Algorithm Based on the Probability of Reference for

Flash Memory. IEEE Access, Vol. 5, 2017, pp. 12626–12634, doi: 10.1109/ACCESS.2017.2723758.

[10] Zhou, W.—Feng, D.—Hua, Y.—Liu, J.—Huang, F.—Chen, Y.—Zhang, S.: Prober: Exploiting Sequential Characteristics in Buffer for Improving SSDs Write Performance. Frontiers of Computer Science, Vol. 10, 2016, No. 5, pp. 951–964, doi: 10.1007/s11704-016-5286-z.

[11] Chen, F.—Koufaty, D. A.—Zhang, X.: Hystor: Making the Best Use of Solid State Drives in High Performance Storage Systems. Proceedings of the International Conference on Supercomputing (ICS '11), ACM, 2011, pp. 22–32, doi: 10.1145/1995896.1995902.

[12] On, S. T.—Gao, S.—He, B.—Wu, M.—Luo, Q.—Xu, J.: FD-Buffer: A Cost-Based Adaptive Buffer Replacement Algorithm for Flash Memory Devices. IEEE Transactions on Computers, Vol. 63, 2014, No. 9, pp. 2288–2301, doi: 10.1109/TC.2013.52.

[13] Li, Z.—Jin, P.—Su, X.—Cui, K.—Yue, L.: CCF-LRU: A New Buffer Replacement Algorithm for Flash Memory. IEEE Transactions on Consumer Electronics, Vol. 55, 2009, No. 3, pp. 1351–1359, doi: 10.1109/TCE.2009.5277999.

[14] Kwon, O.—Lee, J.—Koh, K.: EF-Greedy: A Novel Garbage Collection Policy for Flash Memory Based Embedded Systems. In: Shi, Y., van Albada, G. D., Dongarra, J., Sloot, P. M. A. (Eds.): Computational Science – ICCS '07. Springer, Berlin, Heidelberg, Lecture Notes in Computer Science, Vol. 4490, 2007, pp. 913–920, doi: 10.1007/978-3-540-72590-9_138.

[15] Consuegra, M. E.—Martinez, W. A.—Narasimhan, G.—Rangaswami, R.—Shao, L.—Vietri, G.: Analyzing Adaptive Cache Replacement Strategies. arXiv preprint arXiv:1503.07624, 2015, 26 pp.

[16] Jung, H.—Shim, H.—Park, S.—Kang, S.—Cha, J.: LRU-WSR: Integration of LRU and Writes Sequence Reordering for Flash Memory. IEEE Transactions on Consumer Electronics, Vol. 54, 2008, No. 3, pp. 1215–1223, doi: 10.1109/TCE.2008.4637609.

[17] Corbató, F. J.: A Paging Experiment with the Multics System. In Honor of Philip M. Morse, MIT Press, Cambridge, Mass., 1969, pp. 217–228.

[18] O'Neil, E. J.—O'Neil, P. E.—Weikum, G.: The LRU-K Page Replacement Algorithm for Database Disk Buffering. ACM SIGMOD Record, Vol. 22, 1993, No. 2, pp. 297–306, doi: 10.1145/170036.170081.

[19] Lee, D.—Choi, J.—Kim, J.-H.—Noh, S. H.—Min, S. L.—Cho, Y.—Kim, C. S.: LRFU: A Spectrum of Policies That Subsumes the Least Recently Used and Least Frequently Used Policies. IEEE Transactions on Computers, Vol. 50, 2001, No. 12, pp. 1352–1361, doi: 10.1109/TC.2001.970573.

[20] Jiang, S.—Zhang, X.: LIRS: An Efficient Low Inter-Reference Recency Set Replacement Policy to Improve Buffer Cache Performance. ACM SIGMETRICS Performance Evaluation Review, Vol. 30, 2002, No. 1, pp. 31–42, doi: 10.1145/511399.511340.

[21] Johnson, T.—Shasha, D.: 2Q: A Low Overhead High Performance Buffer Management Replacement Algorithm. Proceedings of the 20th International Conference on Very Large Data Bases (VLDB '94), 1994, pp. 439–450.

[22] Bansal, S.—Modha, D. S.: CAR: Clock with Adaptive Replacement. Proceedings of the 3rd USENIX Conference on File and Storage Technologies (FAST '04), 2004, pp. 187–200.

[23] Su, X.—Jin, P.—Xiang, X.—Cui, K.—Yue, L.: Flash-DBSim: A Simulation Tool for Evaluating Flash-Based Database Algorithms. 2009 2nd IEEE International Conference on Computer Science and Information Technology (ICCSIT '09), Beijing, China, 2009, doi: 10.1109/ICCSIT.2009.5234967.

[24] Zhou, Y.—Chen, Z.—Li, K.: Second-Level Buffer Cache Management. IEEE Transactions on Parallel and Distributed Systems, Vol. 15, 2004, No. 6, pp. 505–519, doi: 10.1109/TPDS.2004.13.

[25] Jung, H.—Yoon, K.—Shim, H.—Park, S.—Kang, S.—Cha, J.: LIRS-WSR: Integration of LIRS and Writes Sequence Reordering for Flash Memory. In: Gervasi, O., Gavrilova, M. L. (Eds.): Computational Science and Its Applications (ICCSA 2007). Springer, Berlin, Heidelberg, Lecture Notes in Computer Science, Vol. 4705, 2007, pp. 224–237, doi: 10.1007/978-3-540-74472-6_18.

[26] Hayes, B.: Cloud Computing. Communications of the ACM, Vol. 51, 2008, No. 7, pp. 9–11, doi: 10.1145/1364782.1364786.

[27] Intel Corporation. Understanding the Flash Translation Layer (FTL) Specification. White Paper, `http://www.embeddedfreebsd.org/Documents/Intel-FTL.pdf`, 1998.

Atul O. Thakare received his post-graduate degree (M.Eng. (CSE)) from SGB Amravati University, Maharashtra, India and his under-graduate degree (B.Eng. (CT)) from RTM Nagpur University, Maharashtra India. Currently, he is pursuing his Ph.D. from VNIT, Nagpur, Maharashtra, India. He has a total of 16 years of experience, six years in the IT industry and ten years in the academic profession.

Parag S. Deshpande has completed his Ph.D. from Nagpur University, Nagpur, India and his M-Tech from IIT Powai, Mumbai, India. He is currently working as Professor in the Department of Computer Science and Engineering, VNIT, Nagpur, Maharashtra, India. He has 31 years of academic experience. He is the author of several books including C and Data Structure, Data Warehousing Using Oracle, SQL/PLSQL for Oracle 11g. He is member of ISTE and SAE-India.

Computing and Informatics, Vol. 38, 2019, 1272–1300, doi: 10.31577/cai_2019_5_1272

CELLULAR AUTOMATA BASED IMAGE AUTHENTICATION SCHEME USING EXTENDED VISUAL CRYPTOGRAPHY

Sonal Kukreja, Geeta Kasana, Singara Singh Kasana

Computer Science and Engineering Department
Thapar Institute of Engineering and Technology
Patiala, Punjab, India, 147004
e-mail: kukreja.sonal@gmail.com, {gkasana, singara}@thapar.edu

Abstract. Most of the Visual Cryptography based image authentication schemes hide the share and authentication data into cover images by using an additional data hiding process. This process increases the computational cost of the schemes. Pixel expansion, meaningless shares and use of codebook are other challenges in these schemes. To overcome these issues, an authentication scheme is proposed in which no embedding into the cover images is performed and meaningful authentication shares are created using the watermark and cover images. This makes the scheme completely imperceptible. The watermark can be retrieved just by superimposing these authentication shares, thus reducing the computational complexity at receiver's side. Cellular Automata is used to construct the master share that provides self-construction ability to the shares. The meaningful authentication shares help in enhancing the security of the scheme while size invariance saves transmission and storage cost. The scheme possesses the ability of tamper detection and its localization. Experimental results demonstrate the improved security and quality of the generated shares of the proposed scheme as compared to existing schemes.

Keywords: Image authentication, cellular automata, extended visual cryptography, watermarking, normalized Hamming similarity, PSNR, SSIM

1 INTRODUCTION

There have been tremendous advancements in data transmission technology in the past decade. The transmitted data can be easily recreated or tampered on the network. Images are popular media used in data transmission. Hence protection and authentication of the images have gained a lot of attention in recent times. Protecting the integrity of image from modification by unauthorized users is called image authentication. Many image authentication schemes [11, 28] have been proposed in the literature that can be classified into two categories:

Digital signature based authentication: In this type of schemes [3, 4, 5, 8, 13, 20, 27, 36, 39, 40], a signature is generated from the image using a hash function. This signature is then embedded into the image. To verify authentication of the image, the signature is again calculated from the image using the same hash function and compared with the extracted signature. If both the signatures are same, the image is ensured to be authenticated else detected as tampered.

Watermarking based authentication: In watermarking based schemes [6, 25, 34, 37, 41], a watermark is embedded into the image. The resultant image is known as the marked image. On the receiver side, the watermark is extracted from the marked image. If the extracted watermark is different from the embedded one, the image is detected to be tampered. The original watermark is required for the comparison.

Apart from watermarking, cryptography also plays a crucial role to protect the data being transmitted. Naor and Shamir [19] proposed a modification of cryptography scheme to share a secret among a number of participants in such a way that every participant gets a part of the secret. This scheme is termed as Visual Cryptography (VC). A (k, n) VC is a scheme in which n random shares are constructed from the secret image and later, it can be reconstructed by superimposing authorized k or more shares where, $k \leq n$. No information can be retrieved from any subset of unauthorized shares or shares less than k. The shares created in this scheme were meaningless in appearance which may create a suspicion of some secret information being shared. Hence, this scheme was modified by Ateniese et al. [12] by creating meaningful shares and termed as Extended Visual Cryptography Scheme (EVCS). Here, when the meaningful shares are superimposed at the time of retrieval, the meaningful information disappears from them and the secret hidden inside the shares is recovered. Both of these schemes suffered from pixel expansion and low contrast in the generated shares.

Kafri and Keren [22] further enhanced the existing VC scheme by handling pixel expansion. Here, the binary secret image S is encoded into two random grids having the same size as that of S. This scheme removed pixel expansion but created random looking meaningless shares. Guo et al. [32] further improved the above scheme to create meaningful random grids. This scheme shows the benefits of random grid based VC to create meaningful shares with no pixel expansion. A probabilistic parameter β is used to make a trade-off of visual quality deterioration between share images and the decoded secret image. The larger value for β results in more visual-pleasing share images and less visual-pleasing decoded image, while smaller value for β results in less visual-pleasing share images and more visual-pleasing decoded image. These schemes have been further enhanced in [43, 42] to improve the visual quality of generated shares.

The existing watermarking based authentication schemes [6, 25, 34, 37, 41] modify the original host image by embedding watermark into it, thereby resulting in the image quality degradation. Hence, in the past decade VC based authentication schemes [3, 5, 8, 13, 30, 36, 39] have gained a lot of attention as these schemes do not embed watermark into the host original image. The shares are constructed from the host image, and then these shares along with the authentication data or watermark are embedded into some other cover images which are transferred through the communication channels. This authentication data is extracted at the receiver's side and used to detect tamper in images. In all these schemes, an additional data hiding scheme is required to hide the constructed shares in some cover images, which increases complexity of these schemes.

The paper has been organized into the following Sections. Section 2 discusses the related works, Section 3 briefly reviews the concepts used in the proposed scheme like Cellular Automata (CA) and Wavelet Packet Decomposition (WPD). Section 4 presents the proposed authentication scheme. Experimental results and discussion are presented in Section 5, followed by conclusions in Section 6.

2 RELATED WORKS

In past few years, many VC based authentication schemes have been proposed that are briefly discussed in this section. In 2004, Lin et al. [5] proposed a VC based authentication scheme in which the shares are generated from the secret image using a polynomial. The image is divided into blocks and parity bits are calculated for every block. These parity bits are used as authentication data. This data together with shares are embedded into cover images. The drawback of this scheme is that the size of the cover image increases to four times the size of the secret image. Yang et al. [8] modified Lin et al. [5] to enhance authentication ability and quality of reconstructed image. This scheme also used polynomial based VC to generate n shares, but unlike the Lin's scheme [5] which set the value of the variable p to 251, this scheme set value of p to the Galois Field (GF), i.e. $p = g(x) = x^8 + x^4 + x^3 + x^1 + x^0$, which reduces the distortion in the received secret image. The extracted secret image has minimum distortion, but the cover image still remains four times the size of secret image. In some cases, this scheme provided fake authentication. Chang et al. [3] proposed another VC based authentication scheme using Chinese Remainder Theorem (CRT) that improved the authentication ability but the issue of pixel expansion still existed. The size of the cover image was relatively decreased as compared to the previous two schemes but it was still twice the size of the secret image. As this scheme uses CRT to evaluate the authentication bits, the computational complexity got increased. Another scheme was proposed by Lou et al. [9] in which the secret image is embedded inside two meaningful cover images, with no pixel expansion. This scheme provides larger embedding capacity at the same transmission cost and better contrast of the generated shares.

The Two in One Secret Sharing Scheme [44] was proposed to provide good quality decoded images while performing decoding without use of any computations. But this scheme had security limitations as it created noisy shares. This scheme was improved by Srividhya et al. [30] by creating meaningful shares and enhancing its security by sharing an authentication image along with the secret image. The visual quality of the recovered image was improved due to usage of adaptive halftoning.

Eslami et al. [39] proposed an embedding scheme where the size of the block depends on the data to be hidden, hence the cover images are used efficiently for hiding the data. This scheme also included an authentication-chaining method which used 2 authentication bits and was able to achieve 15/16 tamper detection ability. But its disadvantage was that after encountering a tampered block, the rest of the image could not be tested. The authentication abilities for the increased block sizes were improved and the individual blocks of the stego image could also be authenticated in Ulutas et al. [13]. The visual quality of the stego images was also better as compared to the exisiting ones and could authenticate the rest of the stego image after encountering an altered block in the stego image.

Another VC scheme with authentication abilities, which could create meaningful shares, using the concepts of cellular automata, DWT and hash functions, was proposed by Wu et al. [36]. This scheme used an additional data hiding process to embed the shares inside cover images. This scheme had low visual quality for stego images and low tampered detection rate.

2.1 Motivation for the Proposed Work

From the study conducted on the existing watermarking based authentication schemes [7, 10, 15, 16, 18, 21, 24, 33, 35, 38], it is observed that the host image is modified to embed the watermark which degrades its visual quality. Also, to retrieve the watermark from the marked image, watermark extraction algorithm is followed, that increases the computation cost of the scheme. Pixel expansion, meaningless shares and use of codebook for share generation remain to be continuing challenges in the existing VC based authentication schemes [3, 4, 5, 8, 30, 36, 39]. These schemes generally require an additional data hiding process to conceal share and authentication bits into images, which increases the computation cost.

2.2 Contribution

The novelty of the proposed scheme is that the original host image is not modified for watermark embedding, instead authentication shares are constructed from host image using VC to store watermark information. This makes the scheme completely imperceptible. As no data hiding process is required to conceal these shares and authentication data into cover images, instead the meaningful authentication shares are constructed that resemble the cover images. This reduces complexity of the scheme as, at the receiver's side, the watermark can be retrieved just by superimposing the shares.

Cellular Automata is used to create the master share, which eliminates pixel expansion and use of codebook, that saves transmission and storage cost. Also, to create meaningful authentication shares, a probabilistic parameter p is taken that decides the tradeoff between visual qualities of authentication shares and extracted watermark. This parameter makes the proposed scheme flexible for real-time applications. Meaningful shares enhance the security of the scheme, as random looking shares create suspicion that some secret information is hidden inside them. Use of VC further enhances security of the scheme, as watermark hidden in the shares cannot be detected or removed easily by any unauthorized entity but is retrieved only when k out of n shares are superimposed with each other.

Also, if an attacker gets access to the image, he can also construct the key and create the master share. The watermark cannot be retrieved until it is superimposed with some authorized shares stored with participants. This ensures the security of the proposed scheme. The key used to create master share can be constructed at both sender and receiver sides. Hence, there is no need to transmit it as side information thereby saving transmission cost.

3 PRELIMINARIES

In this section, Cellular Automata and Wavelet Packet Decomposition are discussed, which are used in the proposed scheme.

3.1 Cellular Automata

Cellular Automata (CA) is an array of entities which are known as cells. Every cell has a finite state having value either 0 or 1. Every cell has a neighborhood, which is usually described by its adjacent cells. The cells of the cellular automata exhibit following properties:

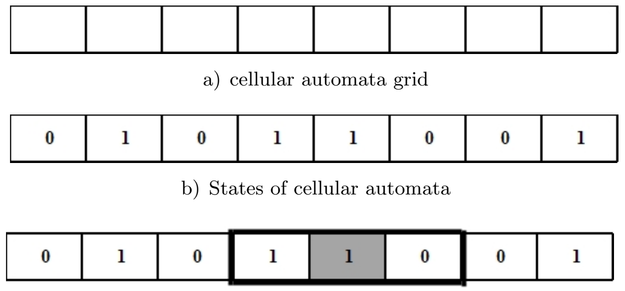
Grid: All cells of cellular automata arrange themselves in the form of a grid, as shown in Figure 1 a).

State: Every cell has a state. The number of state possibilities is typically finite. Every cell usually has 2 states: (0 and 1) or (ON and OFF) or (ALIVE and DEAD), as shown in Figure 1 b).

Neighborhood: Neighborhood involves the cell and its adjacent cells, as shown in Figure 1 c).

The state of the cell at an instant of time t, depends on its state at time $t-1$ along with the state of its neighbors at time $t-1$. A certain set of rules is followed to determine this current state of the cell on the basis of the previous states of the cell and its neighbors. This can be written as:

$$S(t) = F(S(t-1))$$



a) cellular automata grid

b) States of cellular automata

c) Neighborhood of a cell

Figure 1. Representation of cellular automata

where $S(t)$ represents state of a cell at time t. Function F is determined by various rules described in [26]. This has been described in Figure 2.

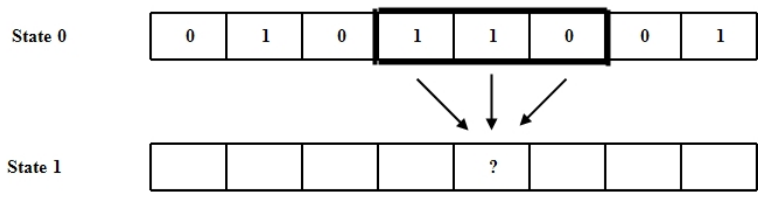


Figure 2. Transition of a cell from one state to the next state

The state configuration for every cell along with its right and left neighborhoods is represented by three bits, e.g. 100. Hence there are total eight possible neighborhood state configurations. Thus the rulesets are represented by eight bits, e.g. Rule 10001010, representing eight different state configurations. In the proposed scheme Rule 30 has been used. Generation of new state from the previous state using Rule 30 is shown in Figure 3.

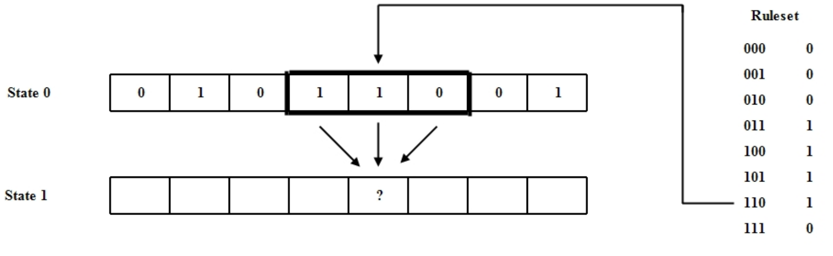


Figure 3. Use of Rule 30 to generate the next state

In terms of a Wolfram elementary CA, there are 256 possible rulesets. The ruleset used here is commonly referred to as Rule 30 because if the binary sequence

00011110 is converted to a decimal number, integer 30 is obtained. The generic CA is extended to two dimensions [26] which permits direct comparisons to real physical systems like crystal growth, chemical reaction-diffusion systems, simulation of turbulent flow patterns, etc. Two Dimensional CA supports variety of lattices and neighborhood structures like von-Neumann neighborhood where the center cell is surrounded by four neighbors, Moore neighborhood that has eight neighbors around the center cell, etc.

3.1.1 Rule 30

Rule 30 is an one dimensional binary CA rule introduced by Stephen Wolfram [31] in 1983. This rule has been used with VC in [26]. As stated in [2], Rule 30 is an exceptional legal rule that is highly periodic and random. Hence, this rule chosen in the proposed scheme helps in enhancing security. It is described in Figure 4. The figure shows all eight possible state configurations and their corresponding next state. The color of the next state is determined by the color of the cell and its neighbors in the previous state. As the binary representation of the outcome of the rule turns out to be 30, the rule is known as Rule 30. ($30 = 00011110_2$)

Figure 4. Representation of different state configurations and their next states using Rule 30

3.2 Wavelet Packet Decomposition

Wavelet Packet Decomposition (WPD), also known as Wavelet Packets, is a wavelet transform where discrete-time sampled signal is passed through more filters as compared to the Discrete Wavelet Transform (DWT) [17, 23]. In DWT, every level is calculated by passing previous wavelet approximation coefficients (low pass results) while in WPD, both detail (high pass results) and approximation coefficients are used to create the full binary tree.

The wavelet decomposition procedure splits the approximation coefficients into two parts. After splitting, we obtain a vector of approximation coefficients and a vector of detail coefficients, both at a coarser scale. The information lost between two successive approximations is captured in the detail coefficients. Then the new approximation coefficient vector is split again. In the wavelet packet approach, each detail coefficient vector is also decomposed into two parts as in approximation vector splitting. A pictorial representation of WPD over three levels is shown in Figure 5.

For n levels of decomposition the WPD produces 2^n different sets of coefficients (or nodes) as opposed to $n+1$ sets for the DWT. However, due to the down sampling process, the overall number of coefficients is still the same and there is no redundancy.

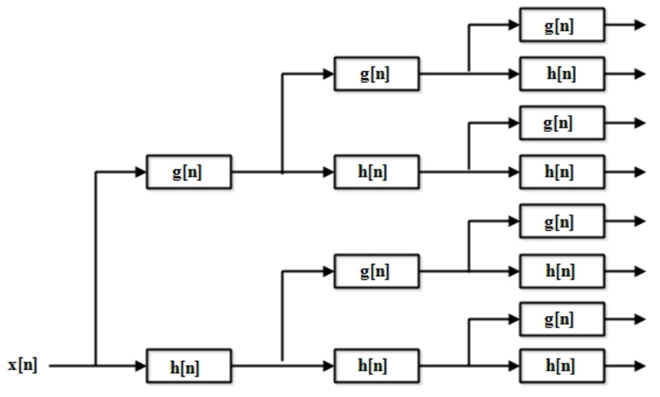


Figure 5. WPD over 3 levels. $g[n]$ is the low-pass approximation coefficients, $h[n]$ is the high-pass detail coefficients [17].

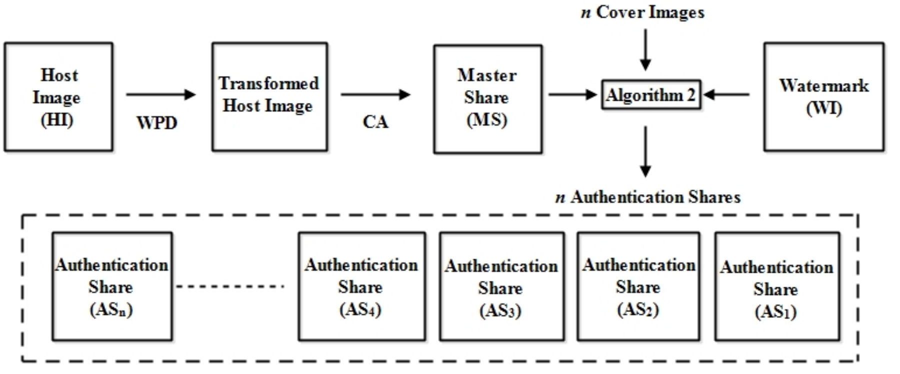
Wavelet Packets have a larger library of functions than wavelets, which help in representing different types of images efficiently. Especially the images that have smaller scale wavelet coefficients and carry very little energy, can be effectively represented by WPD. Thus, in the proposed scheme we have utilized WPD to create a master share that contains maximum features of the image, else it might lead to increase in false positives and false negatives in the tamper detection cases.

4 PROPOSED SCHEME

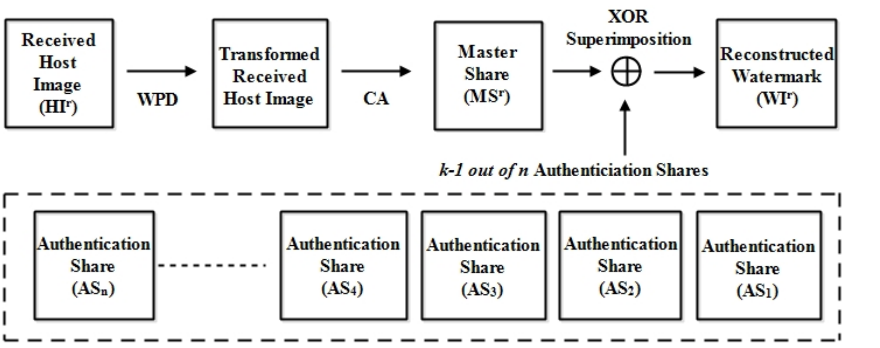
The proposed scheme consists of two phases: Share Generation Phase and Authentication Phase. It has been represented with the help of a block diagram shown in Figure 6.

4.1 Share Generation Phase

In the Share Generation Phase, a share is generated from the host image using WPD and CA. This share is referred as Master Share *(MS)*. This phase is described in Algorithm 1. The Host Image is divided into equal sized blocks. A basis set is produced for every block by applying WPD. The binary code for the average value of this set is considered as the key for the respective block. The obtained key is used to generate the corresponding *MS* block using CA with Rule 30. Generation of *MS* has been explained with the help of an example in Figures 7 and 8. In Figure 7, a best basis set is constructed after applying WPD on an image block. The average of this set is calculated as 17. The binary conversion of 17 is considered as the key for CA, which is used to construct *MS* in Figure 8. The first row of the block is



a) Share generation phase



b) Authentication phase

Figure 6. Block diagram of proposed scheme

initialized with the key and the remaining rows are constructed from this first row using Rule 30.

The benefits of using CA are that no codebook is required to create shares, no pixel expansion as every host image bit is represented by one bit in *MS* and no need to store the share as it can construct itself from its initial state. Existing VC scheme based on CA [26] uses a key to generate the share. This key has to be transmitted from the sender to receiver as side information which results in additional transmission cost and can be accessed by third-party attackers too. While in the proposed scheme as a binary representation of mean for every block is considered as the key, there is no need to transmit the key as side information. It can be constructed by the sender and receiver individually from their host and received image respectively. This reduces transmission cost and enhances security too.

After constructing *MS*, n meaningful and non-expanded authentication shares are generated using *MS* and watermark. Generation of authentication shares is illustrated in Figure 9. For every pixel, a random bit x is generated with probability p.

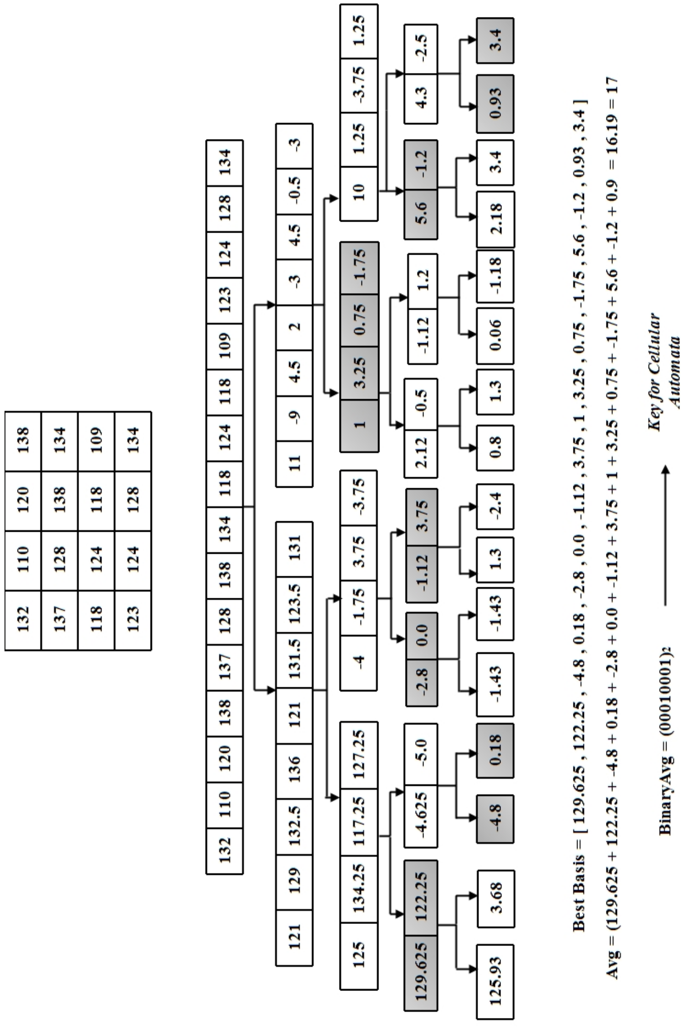


Figure 7. Example of WPD on SI Block of size 4×4

If x turns out to be 0, pixel from watermark is included in the authentication share and if it turns out to be 1, pixel from cover image is included in the authentication share.

In proposed scheme, parameter p is set as a trade-off between visual quality of share images and retrieved Watermark Image (WI), i.e., the smaller value for p results into more visual-pleasing share images and lower image quality of retrieved watermark image, while the larger value for p results in less visual-pleasing share images and higher image quality of retrieved watermark image. The generated

Key →

0	0	0	1	0	0	0	1
1	0	1	1	1	0	1	1
0	0	1	0	0	0	1	0
0	1	1	1	0	1	1	1
0	1	0	0	0	1	0	0
1	1	1	0	1	1	1	0
1	0	0	0	1	0	0	0
1	1	0	1	1	1	0	1

Master Share Block generated using CA

Rule 30

Previous State of cell and its neighbours	Next State of Cell
0 0 0	0
0 0 1	1
0 1 0	1
0 1 1	1
1 0 0	1
1 0 1	0
1 1 0	0
1 1 1	0

Figure 8. Generation of KS

Algorithm 1 Master Share Construction

Input: Host Image (HI) of size $r \times c$
Output: Master Share (MS)
Divide HI into equal size blocks HI^b of size $m \times m$, where $1 \le b \le a$, $a = \frac{r \times c}{m \times m}$
for every block (HI^b) **do**:
  Apply WPD on HI^b and produce a best basis set.
  Calculate average value Avg_b for the basis set
  Convert Avg_b into 8 bit binary code $BinaryAvg_b$
  Create corresponding Master Share by applying Rule 30 on $BinaryAvg_b$
Return *MS*

authentication shares $[AS_1, AS_2, \ldots, AS_n]$ are distributed among n participants and watermark is stored with a Trusting Authority (TA).

4.2 Authentication Phase

In this phase, the images are verified at the receiver's side, whether they have been maliciously tampered. For the received host image HI^r, master share MS^r is constructed using Algorithm 1 and authentication shares are retrieved from $k-1$ participants. These $k-1$ shares are superimposed with MS^r to retrieve the watermark WI^r using $k-1$ operations. The superimposed result WI^r should be similar to the *WI* stored with the *TA*. This similarity is compared using *NHS*, which is defined as:

$$\mathrm{NHS} = \frac{\mathrm{HD}(\mathrm{WI}, \mathrm{WI}^r)}{r \times c} \tag{1}$$

where HD is the Hamming distance between two binary images, WI is original watermark image, WI^r is extracted watermark image and $r \times c$ is the watermark

Algorithm 2 Authentication Shares Generation

Input: Master Share (MS), Watermark Image (WI), n Cover Images $[CI_1, CI_2, \ldots, CI_n]$ of size $r \times c$

Output: n Authentication Shares $[AS_1, AS_2, \ldots, AS_n]$

```
for i = 1 to r do
    for j = 1 to c do
        Generate a bit x, such that x = 1 with probability p and x = 0 with probability
1 − p.
        if x == 1 then
[AS_1(i, j), AS_2(i, j), ..., AS_n(i, j)] =
generateSecretBits(WI(i, j), MS(i, j))
        else
[AS_1(i, j), AS_2(i, j), ..., AS_n(i, j)] =
generateCoverBits(CI_1(i, j), CI_2(i, j), ..., CI_n(i, j))
function GENERATESECRETBITS(w, k, n)
    if w == 0 then
m_1 = k
    else
m_1 = Complement(k)
    for z in range(2, n) do
        if w == 0 then
m_z = m_{z−1}
        else
m_z = Complement(m_{z−1})
    Return [m_1, m_2, ..., m_n]
function GENERATECOVERBITS(c_1, c_2, ..., c_n, n)
    for doz in range(1, n)
        if c_z == 0 then
m_z = 0 or 1
        else
m_z = 1
    Return [m_1, m_2, ..., m_n]
```

image size. If the value of NHS tends towards unity, then original and extracted watermarks are identical and host image is authentic otherwise it is tampered. This phase is shown in Figure 10.

An additional refinement process is applied to WI^r to enhance the results of tamper detection. Logical NOR operation is applied between WI and WI^r for the areas which have been detected as tampered while the rest of the pixels remain unmodified. This helps in enhancing the accuracy of tamper detection scheme.

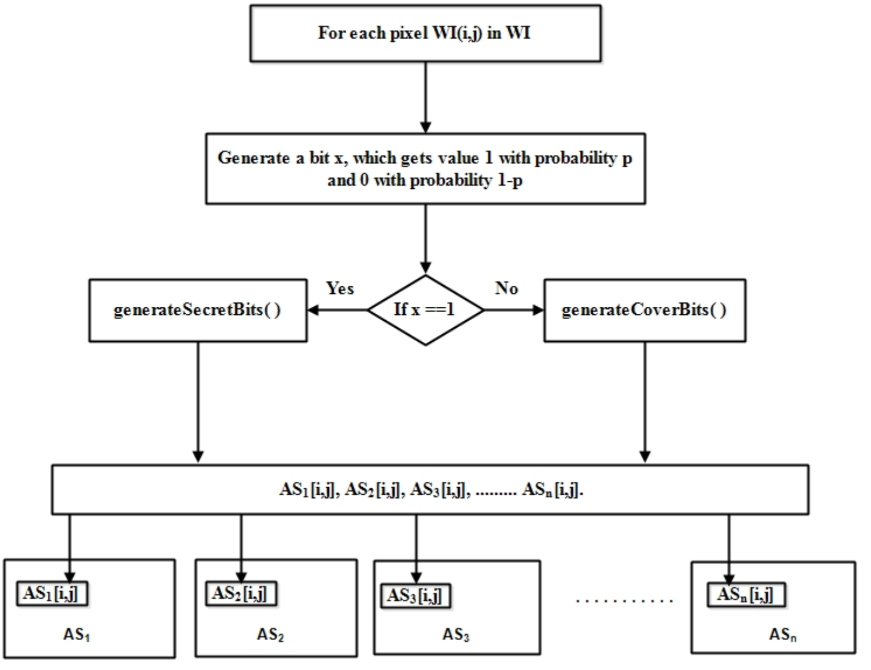


Figure 9. Generation of authentication shares

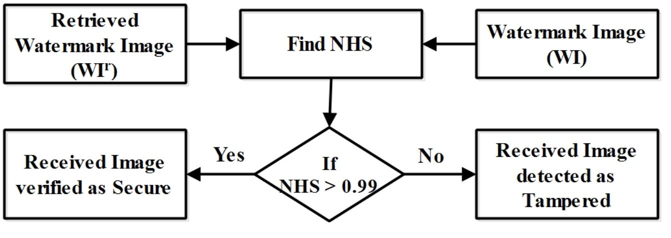
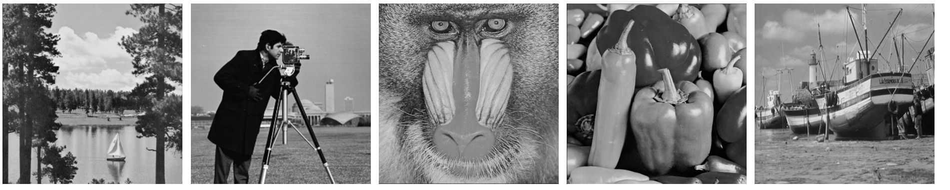


Figure 10. Tamper detection phase

5 EXPERIMENTAL RESULTS AND DISCUSSION

The performance of the proposed scheme is implemented using MATLAB (R2018a), 64-bit (win64) software. The experiment is conducted on 8-bit host images and binary watermark image of size 512×512. The size of the watermark can be lesser than size of Host Image, but that would lead to pixel expansion while generating shares. Five test images *viz.* Lake, Cameraman, Baboon, Peppers, and Boat are used for experimentation purpose and presented in Figure 11.

To evaluate the effectiveness of the proposed authentication scheme, some standard measures viz. Peak Signal to Noise Ratio (PSNR), Structural Similarity Index



a) b) c) d) e)

Figure 11. Different input test images used in the proposed method, a) Lake, b) Cameraman, c) Baboon, d) Peppers, e) Boat (Image source: `http://sipi.usc.edu/database/database.php?volume=misc`)

Module (SSIM) and Tamper Detection Rate (TDR) are used. Statistical analysis of the scheme is also performed using parameters like True Positive Rate (TPR), False Positive Rate (FPR) and accuracy. The efficiency of the proposed scheme is also tested against different tampering attacks. These measures and results are described in the following subsections.

5.1 Quality Analysis

The parameters used to analyze the visual quality of generated authentication shares are PSNR and SSIM while NHS is used to analyze similarity between original WI and retrieved WI^r.

5.1.1 Peak Signal to Noise Ratio

The visual quality of the authentication shares is evaluated using PSNR. It can be defined as:

$$\mathrm{PSNR} = 10 \log \frac{(2^{bd} - 1)^2}{\mathrm{MSE}} \tag{2}$$

where bd is bit depth of the image, MSE represents Mean Square Error between original cover image and authentication share. High value for PSNR shows better quality of the authentication share and least distorted. PSNR of authentication shares in the proposed scheme has been maintained above 50 dB, which is quite better with respect to the existing authentication schemes based on VC. This comparison has been shown in Table 3. PSNR of marked image with respect to original image tends toward infinity as watermark is not embedded into it.

5.1.2 Structural Similarity

This parameter is used to measure the similarity between two images by calculating similarity for various windows of the image. The similarity measure between two windows of same size is given by:

$$\mathrm{SSIM}(x,y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \tag{3}$$

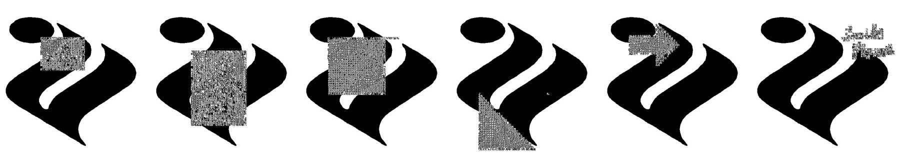
where x and y are the two different windows, μ_x and μ_y are the average values of x and y, σ_x^2 and σ_y^2 are the variance of x and y, σ_{xy} is the covariance of x and y, c_1 and c_2 are two variables to stabilize the division where $c_1 = (k_1 \times L)^2$, $c_2 = (k_2 \times L)^2$, L is the dynamic range of the pixels, $k_1 \leq 1$ is a small constant value.

Results for $(k = 2, n = 3)$-case of the scheme at successive values of $p = 0$, 0.2, 0.4, 0.6, 0.8, 1.0 have been shown in Table 1 (v)–(xviii). $(2, 3)$ represents that 3 shares are generated from the host image and at least 2 shares are required to retrieve the watermark. The table shows KS and MS at different values of p.

It can be observed from Table 1 that as the value of p increases, the visual quality of authentication share images enhances while superimposed result image deteriorates. Thus, depending upon the application and the requirements, the value of p can be chosen, i.e., when the security of the shares being stored is the main concern, a larger value of p would be preferred while if the quality of the watermark retrieved is the main concern to verify the image authentication, smaller value of p can be chosen.

The performance of the proposed scheme is tested against different tampering attacks on all test images. The tampered images and their detection results are shown in Figures 12, 13, 14, 15.

a) Attack 1 b) Attack 2 c) Attack 3 d) Attack 4 e) Attack 5 f) Attack 6



g) Tamper detection for attack 1 h) Tamper detection for attack 2 i) Tamper detection for attack 3 j) Tamper detection for attack 4 k) Tamper detection for attack 5 l) Tamper detection for attack 6

Figure 12. Different tampering attacks and their tamper detection results for Cameraman

It can be observed that the tampered areas in the original cameraman image shown in Figures 12 a)–12 f), are visible as gray patches on the retrieved watermarks

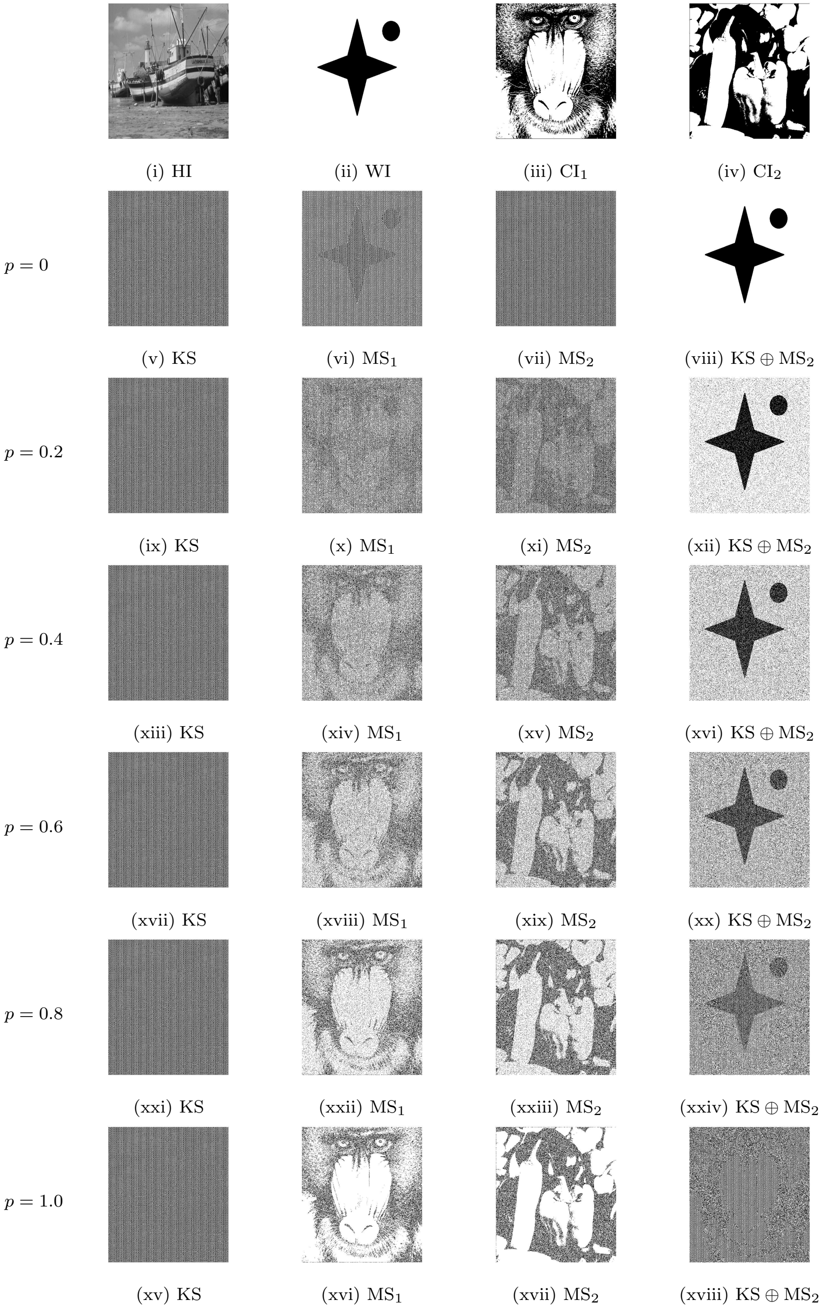
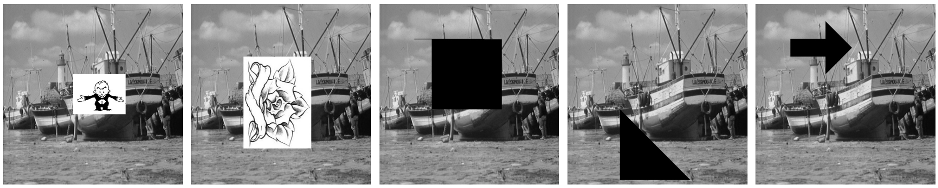
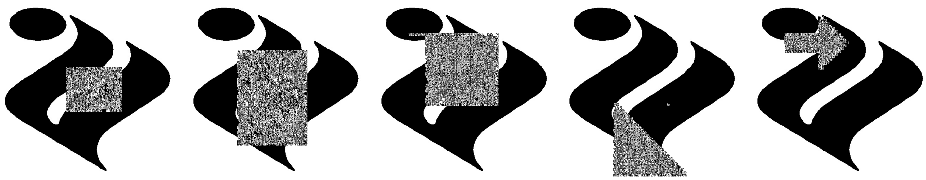


Table 1. Simulation results by proposed scheme for $(k = 2, n = 3)$

in Figures 12 g)–12 l). Similarly, the tamper detection results for Boat, Peppers and Baboon against different tampered attacks are shown in Figures 13, 14, 15.



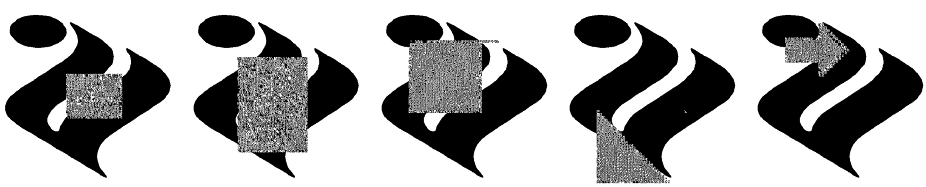
a) Attack 1 b) Attack 2 c) Attack 3 d) Attack 4 e) Attack 5



f) Tamper detection for attack 1 g) Tamper detection for attack 2 h) Tamper detection for attack 3 i) Tamper detection for attack 4 j) Tamper detection for attack 5

Figure 13. Different tampering attacks and their tamper detection results for Boat

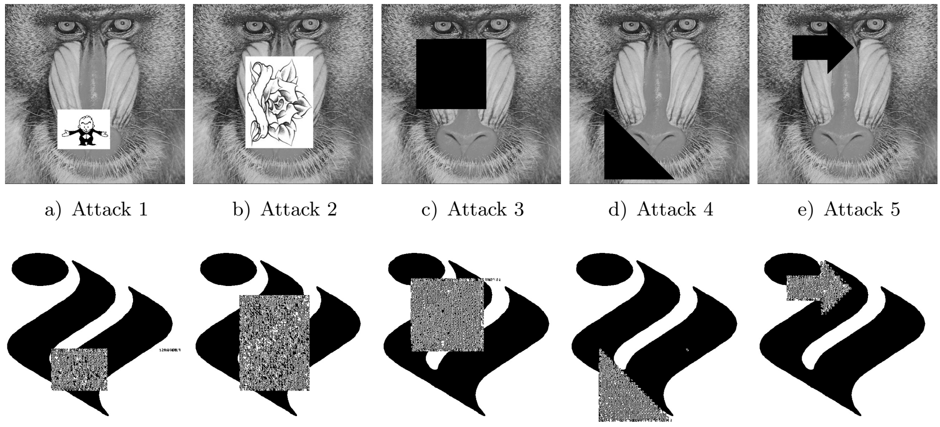
a) Attack 1 b) Attack 2 c) Attack 3 d) Attack 4 e) Attack 5



f) Tamper detection for attack 1 g) Tamper detection for attack 2 h) Tamper detection for attack 3 i) Tamper detection for attack 4 j) Tamper detection for attack 5

Figure 14. Different tampering attacks and their tamper detection results for Peppers

Thus, Figures 12, 13, 14, 15 verify the effectiveness of the proposed scheme in terms of tamper detection. The areas tampered in the attacked images can be visually observed in the retrieved watermark as a gray colored patch.



a) Attack 1 b) Attack 2 c) Attack 3 d) Attack 4 e) Attack 5

f) Tamper detection for attack 1 g) Tamper detection for attack 2 h) Tamper detection for attack 3 i) Tamper detection for attack 4 j) Tamper detection for attack 5

Figure 15. Different tampering attacks and their tamper detection results for Baboon

Table 2 shows the PSNR and SSIM values for the authentication shares generated at different values of p. The results have been shown for AS_1 and AS_2 which are two authentication shares created using the cover images Baboon and Peppers, as shown in Figure 1.

Probability (p)	PSNR		SSIM	
	MS_2	MS_3	MS_2	MS_3
0	54.92	54.29	0.9836	0.9806
0.25	53.04	53.26	0.9801	0.9772
0.5	52.12	52.42	0.9754	0.9741
0.75	51.53	51.74	0.97	0.97
1	51.14	51.16	0.965	0.964

Table 2. PSNR and SSIM of the authentication shares generated for different values of probability (p)

It can be observed from the results that the PSNR of the generated authentication shares have been maintained above 50 dB and SSIM between the share images and the cover images have been maintained close to 1. The values of PSNR and SSIM decrease, as the values of p increase. Table 3 shows the comparison of the PSNR values of the authentication shares of the proposed scheme with the stego images of the existing schemes, that have been created to authenticate the host image. The results for the proposed scheme are shown for an average p value, i.e. $p = 0.5$.

It can be observed from Table 3 that the *PSNR* of these images in the proposed scheme is maintained to be better as compared to the existing schemes. In the

Schemes → Images ↓	Lin et al. [5]	Chang et al. [3]	Chang et al. [4]	Eslami et al. [39]	Yang et al. [8]	Wu et al. [36]	Ulutas et al. [13]	Peng et al. [38]	Proposed Scheme
Baboon	39.18	40.93	45.10	48.10	36.20	47.17	–	40.71	52.12
Lena	39.18	40.97	45.12	48.13	36.17	47.19	48.45	40.73	52.25
Pepper	39.16	40.96	45.1	48.12	36.18	47.18	–	40.72	52.42

Table 3. Comparison with the existing schemes for the PSNR of the authentication shares generated

existing schemes, the stego images are created and stored with hash data and shares embedded into them. While, in the proposed scheme, instead of embedding any data, authentication shares are created using the master share, watermark and cover images.

5.2 Statistical Analysis

Some additional parameters are used to test the effectiveness of the proposed scheme for its tamper detection ability. These parameters are described in Table 4. Two cases have been considered for this evaluation:

Case A: In this case, tamper detection efficiency is analyzed in terms of number of pixels in the image.

Case B: In this case, tamper detection efficiency is analyzed in terms of number of blocks in the image.

5.3 Tamper Detection Rate

Tamper detection rate (TDR) for the scheme for both the cases is shown in Table 10. It can be observed that, for Case A, the detection rate is satisfactory while for Case B it is quite high. This can be defined as: $\mathrm{TDR} = \frac{\mathrm{TP}}{X_{pixels}}$ for Case A, while for Case B, it is defined as: $\mathrm{TDR} = \frac{\mathrm{TP}}{X_{blocks}}$. Value for TP would depend on the Case being followed.

The results of the proposed scheme for the statistical parameters are shown in Tables 3, 4, 5, 6, 7, 8, and 9. Values in these tables show results obtained for various test images at different attacks. Referring to these tables, results clearly demonstrate that the algorithm shows high accuracy in tamper detection. It can be observed that for the blocks, accuracy is around 100 % for most of the images at different attacks. For the pixels it is low, as compared to blocks, but still it is quite high.

Table 10 shows the tampered detection rate of both cases for different images and at different attacks. It can be observed that the proposed scheme shows very high tampered detection rate for Case B while it is satisfactory for Case A.

Table 11 shows the timing analysis of the scheme for different images. It can be observed from this table that the scheme is quite efficient in terms of the time complexity. The construction of authentication shares takes less than one second.

Parameter	Description
T_{pixels}	Total number of pixels in HI^r
X_{pixels}	Total number of tampered pixels in HI^r
T_{blocks}	Total number of blocks in HI^r
X_{blocks}	Total number of tampered blocks in HI^r
True Positive (TP)	A true positive is an outcome where the model correctly predicts the positive class, i.e. number of tampered pixels or blocks that are accurately identified as tampered.
True Negative (TN)	A true negative is an outcome where the model correctly predicts the negative class, i.e. number of untampered pixels or blocks that are accurately identified as untampered.
False Positive (FP)	A false positive is an outcome where the model incorrectly predicts the positive class, i.e. number of pixels or blocks that are tampered but falsely identified as untampered
False Negative (FN)	A false negative is an outcome where the model incorrectly predicts the negative class, i.e. number of pixels or blocks that are untampered but falsely identified as tampered
False Positive Rate (FPR)	$\frac{\mathrm{FP}}{\mathrm{FP}+\mathrm{TN}}$
True Positive Rate (TPR)	$\frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}}$
Accuracy (%)	$\frac{(\mathrm{TP}+\mathrm{TN})}{(\text{Total number of pixels or blocks})}$

Table 4. Parameters used for statistical analysis of tamper detection

Table 12 shows the comparison of the proposed scheme with existing image authentication schemes. In this table, the first column shows different authentication schemes. Subsequently, performance evaluation matrices such as *PSNR*, the similarity between extracted and embedded watermark and tamper detection ability are compared. It can be observed that, as the scheme uses VC, the watermark is not embedded inside the host image, but is hidden in the shares generated. Hence *PSNR* of the marked image with respect to original image tends toward infinity. Also, as XOR operation is used to superimpose *MS* and *AS*, which ensures maximum similarity between the original and extracted watermark, *NHS* tends towards 1. Methods suggested by other authors embed the watermark into the host image, hence the *PSNR* value decreases. Tamper detection ability of the proposed scheme is also high which has been shown in the results. The '–' in the table shows the corresponding data is not available in respective papers.

Table 13 shows the comparison of the proposed scheme with existing image authentication schemes based on VC. In the proposed scheme, CA is used to construct shares and watermark is hidden in these shares. Hence, unlike the existing schemes, there is no requirement of embedding watermark inside the host image and can be

Attack	T_{pixels}		X_{pixels}		TP		FP		TN		FN		TPR		FPR		Accuracy(%)	
Index	Pixels	Blocks	Pixels	Blocks	Pixels	Blocks	Pixels	Blocks	Pixels	Blocks	Pixels	Blocks	Pixels	Blocks	Pixels	Blocks	Pixels	Blocks
1	262 144	4 096	17 097	300	10 595	299	6 502	1	245 047	3 796	989	0	0.9146	1	0.0258	0.00026	97.52	99.98
2	262 144	4 096	50 321	849	32 397	848	17 924	1	211 823	3 247	1 907	0	0.9444	1	0.0780	0.00037	93.16	99.98
3	262 144	4 096	40 000	682	26 763	680	13 237	2	222 144	3 414	1 738	0	0.9390	1	0.0562	0.00058	94.95	99.95
4	262 144	4 096	20 100	377	15 700	377	4 400	0	242 044	3 719	1 931	0	0.8905	1	0.0179	0	98.32	100
5	262 144	4 096	10 700	203	8 025	203	2 675	0	251 444	3 893	1 136	0	0.8760	1	0.0105	0	98.98	100

Table 5. Statistical analysis of tamper detection capacity in terms of pixels and blocks for Boat

Attack	T_{pixels}		X_{pixels}		TP		FP		TN		FN		TPR		FPR		Accuracy(%)	
Index	Pixels	Blocks	Pixels	Blocks	Pixels	Blocks	Pixels	Blocks	Pixels	Blocks	Pixels	Blocks	Pixels	Blocks	Pixels	Blocks	Pixels	Blocks
1	262 144	4 096	17 012	308	10 010	308	7 002	0	245 132	3 788	1 308	0	0.8844	1	0.0278	0	97.33	100
2	262 144	4 096	50 332	850	32 898	850	17 434	0	211 812	3 246	2 016	0	0.9423	1	0.0760	0	93.35	100
3	262 144	4 096	40 000	682	26 546	682	13 454	0	222 144	3 414	1 732	0	0.9388	1	0.0571	0	94.87	100
4	262 144	4 096	20 100	352	17 319	352	2 781	0	242 044	3 414	1 185	0	0.9360	1	0.0114	0	98.94	100
5	262 144	4 096	12 700	227	9 579	227	3 121	0	249 444	3 869	790	0	0.9238	1	0.0124	0	98.81	100

Table 6. Statistical analysis of tamper detection capacity in terms of pixels and blocks for Baboon

Attack	T_{pixels}		X_{pixels}		TP		FP		TN		FN		TPR		FPR		Accuracy(%)	
Index	Pixels	Blocks	Pixels	Blocks	Pixels	Blocks	Pixels	Blocks	Pixels	Blocks	Pixels	Blocks	Pixels	Blocks	Pixels	Blocks	Pixels	Blocks
1	262 144	4 096	17 088	300	10 515	295	6 573	5	245 056	3 796	991	0	0.9139	1	0.0269	0.0013	97.49	99.88
2	262 144	4 096	50 329	849	31 788	841	18 541	8	211 815	3 247	1 943	0	0.9424	1	0.0805	0.0025	92.93	99.80
3	262 144	4 096	39 970	682	26 549	680	13 421	2	222 174	3 414	1 543	0	0.9451	1	0.0570	0.0005	94.88	99.95
4	262 144	4 096	20 092	352	17 289	350	2 803	2	242 052	3 744	1 027	0	0.9439	1	0.0114	0.0005	98.93	99.95
5	262 144	4 096	12 697	227	9 748	223	2 949	4	249 447	3 869	712	0	0.9319	1	0.0117	0.0010	98.88	99.90

Table 7. Statistical analysis of tamper detection capacity in terms of pixels and blocks for Cameraman

Attack	T_{pixels}		X_{pixels}		TP		FP		TN		FN		TPR		FPR		Accuracy(%)	
Index	Pixels	Blocks	Pixels	Blocks	Pixels	Blocks	Pixels	Blocks	Pixels	Blocks	Pixels	Blocks	Pixels	Blocks	Pixels	Blocks	Pixels	Blocks
1	262 144	4 096	17 099	319	11 556	319	5 543	1	245 045	3 777	1 575	0	0.8801	1	0.0221	0	97.89	100
2	262 144	4 096	50 313	849	32 226	848	18 087	1	211 831	3 247	1 921	0	0.9437	1	0.0787	0.0003	93.10	99.98
3	262 144	4 096	40 000	680	26 464	680	13 536	2	222 144	3 414	1 639	0	0.9417	1	0.0574	0.0005	94.84	99.95
4	262 144	4 096	20 100	352	17 300	350	2 800	2	242 044	3 744	1 238	0	0.9332	1	0.0114	0.0005	98.93	99.95
5	262 144	4 096	12 700	224	9 746	224	2 954	3	249 444	3 869	838	0	0.9208	1	0.0117	0.0007	98.87	99.93

Table 8. Statistical analysis of tamper detection capacity in terms of pixels and blocks for Peppers

Attack	T_{pixels}		X_{pixels}		TP		FP		TN		FN		TPR		FPR		Accuracy(%)	
Index	Pixels	Blocks	Pixels	Blocks	Pixels	Blocks	Pixels	Blocks	Pixels	Blocks	Pixels	Blocks	Pixels	Blocks	Pixels	Blocks	Pixels	Blocks
1	262 144	4 096	17 097	300	9 992	299	7 105	1	245 047	3 796	1 028	0	0.9067	1	0.0282	0.0002	97.29	99.98
2	262 144	4 096	50 304	849	32 453	848	17 851	1	211 840	3 247	1 919	0	0.9442	1	0.0777	0.0003	93.19	99.98
3	262 144	4 096	40 000	682	26 517	682	13 483	0	222 144	3 414	1 657	0	0.9412	1	0.0572	0	94.86	100
4	262 144	4 096	20 100	352	17 270	352	2 830	0	242 044	3 744	1 115	0	0.9394	1	0.0116	0	98.92	100
5	262 144	4 096	12 700	227	9 571	227	3 129	0	249 444	3 869	877	0	0.9161	1	0.0124	0	98.81	100

Table 9. Statistical analysis of tamper detection capacity in terms of pixels and blocks for Lake

Images	Boat		Baboon		Lena		Lake		Peppers		Cameraman	
Attack Index	Case A	Case B	Case A	Case B	Case A	Case B	Case A	Case B	Case A	Case B	Case A	Case B
1	66.91	99.67	58.84	100	67.48	100	58.44	99.67	67.58	100	61.53	98.33
2	64.38	99.88	65.36	100	64.57	99.41	64.51	99.88	64.05	99.88	63.16	93.06
3	66.91	99.71	66.36	100	61.25	100	66.29	100	66.16	99.71	66.42	99.71
4	78.10	100	86.16	100	85.93	100	85.92	100	86.07	99.43	86.05	99.43
5	75.00	100	75.43	100	75.55	100	75.36	100	76.74	98.68	76.77	98.24

Table 10. Detection rate for different images at different attacks

extracted just by overlapping the shares without any use of complex extraction algorithm. This reduces the complexity of the scheme. As mean of every block is used as a key, there is no need of transmitting any side information, as mean can be calculated by the sender and receiver individually. This saves the transmission cost. In the existing schemes, all the shares constructed have to be stored with the recipients which are later superimposed together to retrieve the watermark, while in the proposed scheme only one share needs to be stored as the other share is auto-generated using CA. All algorithms can locate tamper detection, but accuracy is either low or not calculated. Most of the reported techniques did not suggest any method to classify attacks quantitatively. While in the proposed scheme, a complete statistical analysis of the scheme has been performed and shown for different images and at different attacks, showing high accuracy.

Images → Algorithms ↓	Boat	Baboon	Lena	Lake	Peppers	Cameraman
Master Share Construction	29.82	36.77	31.84	34.50	30.61	31.80
Authentication Shares Construction	0.64	0.65	0.59	0.61	0.59	0.83
Authentication Phase	46.25	49.55	43.52	46.28	43.72	46.05

Table 11. Timing analysis for different images (in seconds)

5.4 Security Analysis

To prove the security of the proposed scheme, it has been analyzed using the following security aspects.

5.4.1 Construction of Shares

Unlike the existing authentication schemes [24, 15, 7, 21, 10, 18, 16] based on watermarking, the watermark is not embedded inside the host image, but it is used to construct shares. This makes it very difficult to detect or recover the watermark from marked image, thereby making the scheme more secure.

5.4.2 Meaningful Shares

The meaningless random looking shares generated in the traditional VC schemes usually create a suspicion that some secret data is being shared which proves to be a security threat. Thus in the proposed scheme meaningful authentication shares have been generated to ensure the security of the proposed scheme.

Technique	Scheme	PSNR (dB)	Similarity Factor	Tamper Detection/ Localizing	Embedding Required
Chuang et al. [16]	VQ Scheme	≈ 34	–	Possible	Yes
Shen et al. [14]	DWT based scheme	≈ 30	NC = 0.98	Not discussed	Yes
Preda et al. [24]	DWT based scheme	≈ 40	–	Possible	Yes
Al et al. [21]	DWT quantization	≈ 41	–	Possible, can detect 8×8 region	Yes
Li et al. [7]	Two Level DWT	≈ 36	NC = 0.8	Possible, localization accuracy medium	Yes
Li et al. [18]	VQ Scheme	≈ 31.3	SSIM = 0.88	Possible	Yes
Shojanazeri et al. [15]	DWT and Zernike moments	≈ 40.9	–	Possible	Yes
Singh et al. [10]	DCT based scheme	≈ 39.3	NC = 0.98	Possible	Yes
Tiwari et al. [1]	Two stage VQ technique	≈ 42	NHS = 1.0	Possible	Yes
Proposed Work	VC based scheme	*Infinite*	NHS $\approx$ 1.0, NC = 1.0, SSIM = 0.9	Possible, accuracy is very high	No

Table 12. Comparison of proposed scheme with recent image authentication schemes

5.4.3 k out of n scheme

The watermark image has been used to ensure the authentication of the shares generated, thereby enhancing the security. The watermark can be revealed only when k participants superimpose their shares including master share generated from host image. No less than k shares have the ability to extract the watermark. This ensures the security of the scheme.

6 CONCLUSION

In this paper, an authentication scheme based on WPD, VC and CA is proposed. The tampered areas are detected just by XOR-superimposition of shares, thus reducing computational complexity. Experimental results and discussions demonstrate the efficiency of the proposed scheme in terms of imperceptibility, extraction of the hidden watermark with minimum complexity, high accuracy in tamper detection, high security due to meaningful shares, low storage cost and low transmission cost. Also, as compared to some existing authentication schemes based on VC, the proposed scheme can directly generate meaningful authentication shares with wa-

Scheme	Scheme Used for Share Creation	Authentication Data	Embedding Required	Extraction Scheme	Pixel Expansion	Storage Cost for Shares	Transmission Cost for Side Information	Accuracy %
Lin et al. [5]	Polynomial based VC	Parity bits	Yes	Extraction and Overlapping	–	Yes	Yes	–
Chang et al. [3]	Polynomial based VC	Chinese Remainder Theorem	Yes	Extraction and Overlapping	–	Yes	Yes	–
Eslami et al. [39]	Polynomial based VC	Hash Function	Yes	Extraction and Overlapping	–	Yes	Yes	–
Wu et al. [36]	Cellular Automata	Hash Function	Yes	Extraction and Overlapping	–	Yes	Yes	–
Ulutas et al. [13]	Polynomial based VC	Hash Function	Yes	Extraction and Overlapping	–	Yes	Yes	–
Shrividhya et al. [30]	Traditional VC	Authentication Image	No	Extraction and Overlapping	–	Yes	Yes	–
Proposed Work	Cellular Automata and Traditional VC	Watermark	No	XOR-Overlapping	No	Partial	No	High

Table 13. Comparison of proposed scheme with existing image authentication schemes based on VC

termark, host and cover image information, without any extra data hiding process. Tamper detection rate and accuracy have been observed more than 99 % for different images against different tamper attacks. The proposed scheme can be extended to color images.

REFERENCES

[1] Tiwari, A.—Sharma, M.—Tamrakar, R. K.: Watermarking Based Image Authentication and Tamper Detection Algorithm Using Vector Quantization Approach. AEU – International Journal of Electronics and Communications, Vol. 78, 2017, pp. 114–123, doi: 10.1016/j.aeue.2017.05.027.

[2] Ilachinski, A.: Cellular Automata – A Discrete Universe. World Scientific, 2001, doi: 10.1142/4702.

[3] Chang, C.-C.—Hsieh, Y.-P.—Lin, C.-H.: Sharing Secrets in Stego Images with Authentication. Pattern Recognition, Vol. 41, 2008, No. 10, pp. 3130–3137, doi: 10.1016/j.patcog.2008.04.006.

[4] Chang, C.-C.—Chen, Y.-H.—Wang, H.-C.: Meaningful Secret Sharing Technique with Authentication and Remedy Abilities. Information Sciences, Vol. 181, 2011, No. 14, pp. 3073–3084, doi: 10.1016/j.ins.2011.03.002.

[5] Lin, C.-C.—Tsai, W.-H.: Secret Image Sharing with Steganography and Authentication. Journal of Systems and Software, Vol. 73, 2004, No. 3, pp. 405–414, doi: 10.1016/S0164-1212(03)00239-5.

[6] Lo, C.-C.—Hu, Y.-C.: A Novel Reversible Image Authentication Scheme for Digital Images. Signal Processing, Vol. 98, 2014, pp. 174–185, doi: 10.1016/j.sigpro.2013.11.028.

[7] Li, C.—Zhang, A.—Liu, Z.—Liao, L.—Huang, D.: Semi-Fragile Self-Recoverable Watermarking Algorithm Based on Wavelet Group Quantization and Double Authentication. Multimedia Tools and Applications, Vol. 74, 2015, No. 23, pp. 10581–10604, doi: 10.1007/s11042-014-2188-7.

[8] Yang, C. N.—Chen, T. S.—Yu, K. H.—Wang, C. C.: Improvements of Image Sharing with Steganography and Authentication. Journal of Systems and Software, Vol. 80, 2007, No. 7, pp. 1070–1076, doi: 10.1016/j.jss.2006.11.022.

[9] Lou, D. C.—Chen, H. H.—Wu, H. C.—Tsai, C. S.: A Novel Authenticatable Color Visual Secret Sharing Scheme Using Non-Expanded Meaningful Shares. Displays, Vol. 32, 2011, No. 3, pp. 118–134, doi: 10.1016/j.displa.2011.02.001.

[10] Singh, D.—Singh, S. K.: DCT Based Efficient Fragile Watermarking Scheme for Image Authentication and Restoration. Multimedia Tools and Applications, Vol. 76, 2017, No. 1, pp. 953–977, doi: 10.1007/s11042-015-3010-x.

[11] Frattolillo, F.: Watermarking Protocols: An Excursus to Motivate a New Approach. International Journal of Information Security, Vol. 17, 2018, No. 5, pp. 587–601, doi: 10.1007/s10207-017-0386-9.

[12] Ateniese, G.—Blundo, C.—De Santis, A.—Stinson, D. R.: Extended Schemes for Visual Cryptography. Theoretical Computer Science, Vol. 250, 2001, pp. 143–161, doi: 10.1016/S0304-3975(99)00127-9.

[13] Ulutas, G.—Ulutas, M.—Nabiyev, V. V.: Secret Image Sharing Scheme with Adaptive Authentication Strength. Pattern Recognition Letters, Vol. 34, 2013, No. 3, pp. 283–291, doi: 10.1016/j.patrec.2012.10.017.

[14] Shen, H.—Chen, B.: From Single Watermark to Dual Watermark: A New Approach for Image Watermarking. Computers and Electrical Engineering, Vol. 38, 2012, No. 5, pp. 1310–1324, doi: 10.1016/j.compeleceng.2011.11.012.

[15] Shojanazeri, H.—Adnan, W. A. W.—Ahmad, S. M. S.—Rahimipour, S.: Authentication of Images Using Zernike Moment Watermarking. Multimedia Tools and Applications, Vol. 76, 2017, No. 1, pp. 577–606, doi: 10.1007/s11042-015-3018-2.

[16] Chuang, J.-C.—Hu, Y.-C.: An Adaptive Image Authentication Scheme for Vector Quantization Compressed Image. Journal of Visual Communication and Image Representation, Vol. 22, 2011, No. 5, pp. 440–449, doi: 10.1016/j.jvcir.2011.03.011.

[17] Jensen, A.—la Cour-Harbo, A.: Ripples in Mathematics: The Discrete Wavelet Transform. Springer Verlag, 2001, doi: 10.1007/978-3-642-56702-5.

[18] Li, M.—Xiao, D.—Zhang, Y.: Attack and Improvement of the Fidelity Preserved Fragile Watermarking of Digital Images. Arabian Journal for Science and Engineering, Vol. 41, 2016, No. 3, pp. 941–950, doi: 10.1007/s13369-015-1941-1.

[19] Naor, M.—Shamir, A.: Visual Cryptography. In: De Santis, A. (Ed.): Advances in Cryptology – EuroCrypt '94. Springer, Berlin, Heidelberg, Lecture Notes in Computer Science, Vol. 950, 1995, pp. 1–12, doi: 10.1007/BFb0053419.

[20] Singh, M.—Kaur, H.—Kakkar, A.: Digital Signature Verification Scheme for Image Authentication. 2015 2nd International Conference on Recent Advances in

Engineering and Computational Sciences (RAECS), Chandigarh, India, 2015, doi: 10.1109/RAECS.2015.7453277.

[21] Al-Otum, H. M.: Semi-Fragile Watermarking for Grayscale Image Authentication and Tamper Detection Based on an Adjusted Expanded-Bit Multiscale Quantization-Based Technique. Journal of Visual Communication and Image Representation, Vol. 25, 2014, No. 5, pp. 1064–1081, doi: 10.1016/j.jvcir.2013.12.017.

[22] Kafri, O.—Keren, E.: Encryption of Pictures and Shapes by Random Grids. Optics Letters, Vol. 12, 1987, No. 6, pp. 377–379, doi: 10.1364/OL.12.000377.

[23] Percival, D. B.—Walden, A. T.: Wavelet Methods for Time Series Analysis. Cambridge University Press, 2000, doi: 10.1017/CBO9780511841040.

[24] Preda, R. O.: Semi-Fragile Watermarking for Image Authentication with Sensitive Tamper Localization in the Wavelet Domain. Measurement, Vol. 46, 2013, No. 1, pp. 367–373, doi: 10.1016/j.measurement.2012.07.010.

[25] Preda, R. O.—Vizireanu, D. N.: Watermarking-Based Image Authentication Robust to JPEG Compression. Electronics Letters, Vol. 51, 2015, No. 23, pp. 1873–1875, doi: 10.1049/el.2015.2522.

[26] Yampolskiy, R. V.—Rebolledo-Mendez, J. D.—Hindi, M. M.: Password Protected Visual Cryptography via Cellular Automaton Rule 30. In: Shi, Y. Q., Liu, F., Yan, W. (Eds.): Transactions on Data Hiding and Multimedia Security IX. Springer, Berlin, Heidelberg, Lecture Notes in Computer Science, Vol. 8363, 2014, pp. 57–67, doi: 10.1007/978-3-642-55046-1_4.

[27] Alam, S.—Jamil, A.—Saldhi, A.—Ahmad, M.: Digital Image Authentication and Encryption Using Digital Signature. International Conference on Advances in Computer Engineering and Applications, Ghaziabad, India, 2015, pp. 332–336, doi: 10.1109/ICACEA.2015.7164725.

[28] Han, S.-H.—Chu, C.-H.: Content-Based Image Authentication: Current Status, Issues, and Challenges. International Journal of Information Security, Vol. 9, 2010, No. 1, pp. 19–32, doi: 10.1007/s10207-009-0093-2.

[29] Shyu, S. J.: Image Encryption by Random Grids. Pattern Recognition, Vol. 40, 2007, No. 3, pp. 1014–1031, doi: 10.1016/j.patcog.2006.02.025.

[30] Srividhya, S.—Sathishjumar, R.—Sudha, G. F.: Implementation of TiOISSS with Meaningful Shadows and with an Additional Authentication Image. Journal of Visual Communication and Image Representation, Vol. 38, 2016, pp. 284–296, doi: 10.1016/j.jvcir.2016.03.012.

[31] Wolfram, S.: Statistical Mechanics of Cellular Automata. Reviews of Modern Physics, Vol. 55, 1983, No. 3, pp. 601–644, doi: 10.1103/RevModPhys.55.601.

[32] Guo, T.—Liu, F.—Wu, C.: K out of K Visual Cryptography Scheme by Random Grids. Signal Processing, Vol. 94, 2014, pp. 90–101, doi: 10.1016/j.sigpro.2013.06.003.

[33] Nguyen, T.-S.—Chang, C.-C.—Yang, X.-Q.: A Reversible Image Authentication Scheme Based on Fragile Watermarking in Discrete Wavelet Transform Domain. AEU – International Journal of Electronics and Communications, Vol. 70, 2016, No. 8, pp. 1055–1061, doi: 10.1016/j.aeue.2016.05.003.

[34] Korzhik, V.—Zhuvikin, A.—Morales-Luna, G.: Selective Image Authentication Tolerant to JPEG Compression. 2015 6$^{\text{th}}$ International Conference on Infor-

mation, Intelligence, Systems and Applications (IISA), Corfu, Greece, 2015, doi: 10.1109/IISA.2015.7388076.

[35] Hong, W.—Chen, M.—Chen, T. S.: An Efficient Reversible Image Authentication Method Using Improved PVO and LSB Substitution Techniques. Signal Processing: Image Communication, Vol. 58, 2017, pp. 111–122, doi: 10.1016/j.image.2017.07.001.

[36] Wu, X.—Sun, W.: Secret Image Sharing Scheme with Authentication and Remedy Abilities Based on Cellular Automata and Discrete Wavelet Transform. The Journal of Systems and Software, Vol. 86, 2013, No. 4, pp. 1068–1088, doi: 10.1016/j.jss.2012.11.021.

[37] He, Y.—Yang, G.—Zhu, N.: A Real-Time Dual Watermarking Algorithm of H.264/AVC Video Stream for Video-on-Demand Service. AEU – International Journal of Electronics and Communications, Vol. 66, 2012, No. 4, pp. 305–312, doi: 10.1016/j.aeue.2011.08.007.

[38] Peng, Y.—Niu, X.—Yin, Z.: Image Authentication Scheme Based on Reversible Fragile Watermarking with Two Images. Journal of Information Security and Applications, Vol. 40, 2018, pp. 236–246, doi: 10.1016/j.jisa.2018.04.007.

[39] Eslami, Z.—Ahmadabadi, J. Z.: Secret Image Sharing with Authentication-Chaining and Dynamic Embedding. Journal of Systems and Software, Vol. 84, 2011, No. 5, pp. 803–809, doi: 10.1016/j.jss.2011.01.002.

[40] Ziaullah, M.—Shetty, P.—Kamal, S.: Image Feature Based Authentication and Digital Signature for Wireless Data Transmission. 2016 International Conference on Computer Communication and Informatics (ICCCI), Coimbatore, India, 2016, doi: 10.1109/ICCCI.2016.7480009.

[41] Yin, Z.—Niu, X.—Zhou, Z.—Tang, J.—Luo, B.: Improved Reversible Image Authentication Scheme. Cognitive Computation, Vol. 8, 2016, No. 5, pp. 890–899, doi: 10.1007/s12559-016-9408-6.

[42] Qin, H. D.—Dai, Y.—Wang, Z.: A Secret Sharing Scheme Based on (t, n) Threshold and Adversary Structure. International Journal of Information Security, Vol. 8, 2009, No. 5, pp. 379–385.

[43] Hadavi, M. A.—Jalili, R.—Damiani, E.—Cimato, S.: Security and Searchability in Secret Sharing-Based Data Outsourcing. International Journal of Information Security, Vol. 14, 2015, No. 6, pp. 513–529, doi: 10.1007/s10207-015-0277-x.

[44] Lin, S.-J.—Lin, J.-C.: VCPSS: A Two-in-One Two-Decoding-Options Image Sharing Method Combining Visual Cryptography (VC) and Polynomial-Style Sharing (PSS) Approaches. Pattern Reognition, Vol. 40, 2007, pp. 3562–3666, doi: 10.1016/j.patcog.2007.04.001.

Sonal Kukreja graduated in 2012 from Rajasthan Technical University, Kota, India and received her M.Tech. degree in 2015 in the field of computer science and applications from the Thapar Institute of Engineering and Technology, Patiala, India. In 2016, she enrolled as doctoral student at the Thapar Institute of Engineering and Technology, Patiala, India. Her research interest has focused on information security, particularly on data hiding in images, visual cryptography and watermarking.

Geeta Kasana is working as Assistant Professor in Computer Science and Engineering Department, Thapar University, Patiala, India. She has twelve years of teaching and research experience. She received her Ph.D. degree in information security from the Thapar Institute of Engineering and Technology. Her research interests include image processing and information security. She has published many research papers in reputed international journals and conferences. She is currently guiding Ph.D. students on information security.

Singara Singh Kasana is working as Associate Professor in Computer Science and Engineering Department, Thapar Institute of Engineering and Technology, Patiala, Punjab, India. He has nineteen years of teaching and research experience. He received his Ph.D. degree in image compression from the Thapar Institute of Engineering and Technology. His research interests include image processing, wireless networks, and information security. He has published many research papers in reputed international journals and conferences. He is currently guiding Ph.D. students on information security, machine learning and remote sensing.

Computing and Informatics, Vol. 38, 2019, 1301–1319, doi: 10.31577/cai_2019_5_1301

IMAGE SUPER-RESOLUTION BASED ON SPARSE CODING WITH MULTI-CLASS DICTIONARIES

Xiuxiu Liao, Kejia Bai, Qian Zhang
Xiping Jia, Shaopeng Liu, Jin Zhan

School of Computer Science
Guangdong Polytechnic Normal University
No. 293, Zhongshan Avenue West, Tianhe District
510665 Guangzhou, China
e-mail: xxliao2013@aliyun.com, baikejia@gmail.com,
{234482377, 260628072, 149265005, 82952836}@qq.com

Abstract. Sparse coding-based single image super-resolution has attracted much interest. In this paper, a super-resolution reconstruction algorithm based on sparse coding with multi-class dictionaries is put forward. We propose a novel method for image patch classification, using the phase congruency information. A sub-dictionary is learned from patches in each category. For a given image patch, the sub-dictionary that belongs to the same category is selected adaptively. Since the given patch has similar pattern with the selected sub-dictionary, it can be better represented. Finally, iterative back-projection is used to enforce global reconstruction constraint. Experiments demonstrate that our approach can produce comparable or even better super-resolution reconstruction results with some existing algorithms, in both subjective visual quality and numerical measures.

Keywords: Image patch classification, multi-class dictionaries, phase congruency, sparse coding, super-resolution

1 INTRODUCTION

Image super-resolution (SR) refers to the problem of using signal processing techniques to estimate a high-resolution (HR) image $\boldsymbol{X}$ with better quality from an observed low-resolution (LR) image $\boldsymbol{Y}$. The image observation model is usually described as [1, 2]:

$$\boldsymbol{Y} = SH\boldsymbol{X} + \boldsymbol{V} \tag{1}$$

where H is a blurring operator, S is a down-sampling operator and $\boldsymbol{V}$ is additive noise.

In recent years, learning-based SR methods [3] have been extensively studied, which use a learned co-occurrence to predict the correspondence between LR and HR patches. The learning algorithms including Markov network [4, 5, 6], neighbor embedding [7, 8, 9, 10], dictionary learning [11, 12, 13, 14], anchored neighborhood regression [16, 15], random forests [17], and deep learning [18, 19, 20, 21].

Freeman et al. [4] propose an approach named VISTA (Vision by Image/Scene TrAining). They generate a synthetic world of scenes and their corresponding rendered images, modeling their relationships with a Markov network. Bayesian belief propagation is used to efficiently find a local maximum of the posterior probability for the scene if an image is given. They apply VISTA to the super resolution problem, such as estimating high frequency details from a low-resolution image. Stephenson and Chen [5] propose to use even stronger prior information by extending Markov random filed (MRF)-based super-resolution to use adaptive observation and transition functions, that is, to make these functions region-dependent. Ma et al. [6] learn the parameters of the network from training set, which computes probability distribution by K-means algorithm. Given a low-resolution image as input, Chang et al. [7] recover its high-resolution counterpart using a set of training examples. Specifically, small image patches in the low and high-resolution images form manifolds with similar local geometry in two distinct feature spaces. As in locally linear embedding (LLE), local geometry is characterized by how a feature vector corresponding to a patch can be reconstructed by its neighbors in the feature space. Besides using the training image pairs to estimate the high-resolution embedding, they also enforce local compatibility and smoothness constraints between patches in the target high-resolution image through overlapping. Zhang et al. [8] propose a partial least squares (PLS) method, called locality preserving PLS (LPPLS), to find a unified feature space where the correlation between LR and HR image patches on that space is maximized. Applying the proposed LPPLS, they learn the joint mapping of LR and HR image patches simultaneously and then map these image patches onto the unified feature space. The k-nearest neighbor (k-NN) searching and the optimal reconstruction weights computing are performed in this unified feature space as well. Rahiman and George [9] propose learning-based approaches for single image super-resolution using sparse representation and neighbor embedding. Separate prediction models are trained for each cluster, and the model parameters are updated with each input image to adapt to input test image. Gao et al. [10] propose a sparse neighbor selection scheme for SR reconstruction. They first predetermine a larger number of neighbors as potential candidates and develop an algorithm to simultaneously find the neighbors and to solve the reconstruction weights. Recognizing that the k-nearest neighbor for reconstruction should have similar local geometric structures based on clustering, they employ a local statistical feature, namely histograms of oriented gradients (HoG) of LR image patches, to perform such clustering. By conveying local structural information of HoG in the synthesis stage, the k-NN of each LR input patch is adaptively chosen from their associated subset, which significantly improves the speed of synthesizing the HR image while preserving the quality of reconstruction. Wang et al. [11] propose a semi-coupled dictionary learning (SCDL) model to solve cross-style image synthesis problems. Under SCDL, a pair of dictionaries and a mapping function will be simultaneously learned. The dictionary pair can well characterize the structural domains of the two styles of images, while the mapping function can reveal the intrinsic relationship between the two styles' domains. The two dictionaries will not be fully coupled so that much flexibility can be given to the mapping function for an accurate conversion across styles. Moreover, clustering and image nonlocal redundancy are introduced to enhance the robustness of SCDL. He et al. [12] apply a Bayesian method using a beta process prior to learn the over-complete dictionaries. They not only provide dictionaries that customized to each feature space, but also add more consistent and accurate mapping between the two feature spaces. The proposed algorithm is able to learn sparse representations that correspond to the same dictionary atoms with the same sparsity but different values in coupled feature spaces, thus bringing consistent and accurate mapping between coupled feature spaces.

Timofte et al. [15] propose fast super-resolution methods while making no compromise on quality. First, they support the use of sparse learned dictionaries in combination with neighbor embedding methods. In this case, the nearest neighbors are computed using the correlation with the dictionary atoms rather than the Euclidean distance. Second, they show that using global collaborative coding has considerable speed advantages, reducing the super-resolution mapping to a precomputed projective matrix. Third, they propose the anchored neighborhood regression (ANR) algorithm to anchor the neighborhood embedding of a low resolution patch to the nearest atom in the dictionary and to precompute the corresponding embedding matrix. In their later work [16], an improved variant of ANR (A+) is proposed which combines the best qualities of ANR and simple functions (SF) [22]. A+ builds on the features and anchored regressors from ANR. Instead of learning the regressors on the dictionary, it uses the full training material, similar to SF. Schulter et al. [17] propose to directly map from low to high-resolution patches using random forests. They demonstrate how random forests nicely fit into this framework. During the process of trees training, they optimize a novel and effective regularized objective that not only operates on the output space but also on the input space, which especially suits the regression task. During inference, they comprise the same well-known computational efficiency that has made random forests popular for many computer vision problems.

Dong et al. [18] propose a deep learning method for single image super-resolution which directly learns an end-to-end mapping between the low/high-resolution images. The mapping is represented as a deep convolutional neural network (CNN) that takes the low-resolution image as the input and outputs the high-resolution one. Their deep CNN has a lightweight structure, which demonstrates state-of-the-art restoration quality, and achieves fast speed for practical on-line usage. They extend the network to cope with three color channels simultaneously, and show better overall reconstruction quality. Kim et al. [19] use a very deep convolutional network inspired by Visual Geometry Group (VGG)-net for ImageNet classification. Lai et al. [20] propose the Laplacian Pyramid Super-Resolution Network (LapSRN) to progressively reconstruct the sub-band residuals of high-resolution images. At each pyramid level, the model takes coarse-resolution feature maps as input, predicts the high-frequency residuals, and uses transposed convolutions for upsampling to the finer level. They train the proposed LapSRN with deep supervision using a robust Charbonnier loss function and achieve high-quality reconstruction. Furthermore, the network generates multi-scale predictions in one feed-forward pass through the progressive reconstruction, thereby facilitates resource-aware applications. Liu et al. [21] argue that domain expertise from the conventional sparse coding model can be combined with the key ingredients of deep learning to achieve further improved results.

Yang et al. [13] propose sparse coding-based SR framework. Image patches are assumed to have a sparse representation with respect to an over-complete dictionary, and the most relevant reconstruction neighbors are adaptively selected based on sparse coding, avoiding under- or over-fitting. Zeyde et al. [14] embark from the work of [13], similarly assume a local Sparse-Land model on image patches serving as regularization, but use a different training approach for the dictionary pair. Both methods aim at learning a universal dictionary. However, the contents may vary significantly across different image patches, and sparse decomposition over a universal dictionary is potentially unstable [23]. Adaptive sparse coding via multiple dictionaries has been proposed. Yang et al. [24] employ multiple dictionaries learned from K-means clustered patches. Dong et al. [25] use principal component analysis (PCA) technique to learn the sub-dictionaries, and autoregressive and nonlocal self-similarity are introduced as regularization terms. These methods do not use the geometric information as a supervised prior to guide the image patch clustering.

In this paper, a novel method for image patch classification is proposed, and it is integrated into the multiple dictionaries learning SR framework, called SR reconstruction based on Sparse Coding with Multi-Class Dictionary (SC-MCD). Employing the Phase Congruency (PC) measurement [26], image patches are divided into non-smooth patches with different orientations and smooth patches. PC provides an absolute measure of the significance of a local structure, and it is invariant to changes in illumination and magnification. The example patches are classified into several categories, and each category consists of patches with similar patterns, where a sub-dictionary can be learned. For an image patch to be coded, the sub-dictionary that belongs to the same category is selected adaptively. Since the given patch has similar pattern with the selected sub-dictionary, it can be better represented, and the whole image can be more accurately reconstructed. Besides, we use iterative back-projection (IBP) [27] to enforce global reconstruction constraint, which is simple but effective.

The remainder of this paper is organized as follows. In Section 2, we present the proposed SC-MCD algorithm in detail. Experimental results are then presented in Section 3. Finally, Section 4 gives some concluding remarks and discussions on future works.

2 PROPOSED APPROACH

2.1 Sparse Representation-Based SR

Following the image observation model (1), the task of SR is to estimate an HR image $\boldsymbol{X}$ satisfying reconstruction constraint, which requires that $\boldsymbol{X}$ should be consistent with the observed LR image $\boldsymbol{Y}$ with respect to (1). It is ill-posed as many HR images satisfy the reconstruction constraint. Sparse representation-based SR methods use sparse prior on local patches to regularize the estimate of HR image.

Let x, y denote the HR and LR image patches, respectively, two dictionaries $\boldsymbol{D}_h$ and $\boldsymbol{D}_l$ are trained to have the same sparse representations for each HR and LR image patch pair. The recovery of x from y under the sparse prior can be described as:

$$\min_{\alpha,x} \|\alpha\|_0 \text{ s.t. } \|F\boldsymbol{D}_l\alpha - Fy\|_2^2 \le \epsilon_1,\ \|F\boldsymbol{D}_h\alpha - Fx\|_2^2 \le \epsilon_2,\ \|SH\boldsymbol{X} - \boldsymbol{Y}\|_2^2 \le \epsilon_3 \tag{2}$$

where F is a (linear) feature extraction operator, α is the sparse representation coefficients, $\|\alpha\|_0$ represents the number of non-zero coefficients in α, and ϵ_i, $i = 1, 2, 3$ are the admissible errors.

2.2 Classification of Image Patches

Learning a universal dictionary able to optimally represent image patches with various patterns is very difficult. So it is meaningful to learn multiple dictionaries with different patterns. In this section, a novel method for image patch classification is proposed, employing the PC information.

Rather than defining features directly at points with sharp changes in intensity, the PC model postulates that features are perceived at points where the Fourier components are maximal in phase. Kovesi [26] proposes to calculate PC with logarithmic Gabor wavelets. PC at location i is expressed as the summation over orientation o and scale n:

$$PC(i) = \frac{\sum_0 \sum_n W_0(i)\lfloor A_{no}(i)\Delta\Phi_{no}(i) - T_0\rfloor}{\sum_0 \sum_n A_{no}(i) + \epsilon} \tag{3}$$

where $\lfloor x \rfloor$ equals to x when $x > 0$ and it equals to 0 when $x < 0$. A represents the amplitude of the Fourier component, and $\Delta\Phi$ is phase deviation. T compensates for the influence of noise, W is the weighting function for frequency spread, and ϵ is a small constant to avoid division by zero.

Once the PC map of the image is obtained, a threshold P_0 is used to get a binary image, where '1' indicates feature points. At the same time, an orientation image is computed, recording the direction angle in which local energy is a maximum for each pixel. The direction angles are uniform sampled in the range $[0°, 180°]$, and the sample interval is determined by the number of orientation.

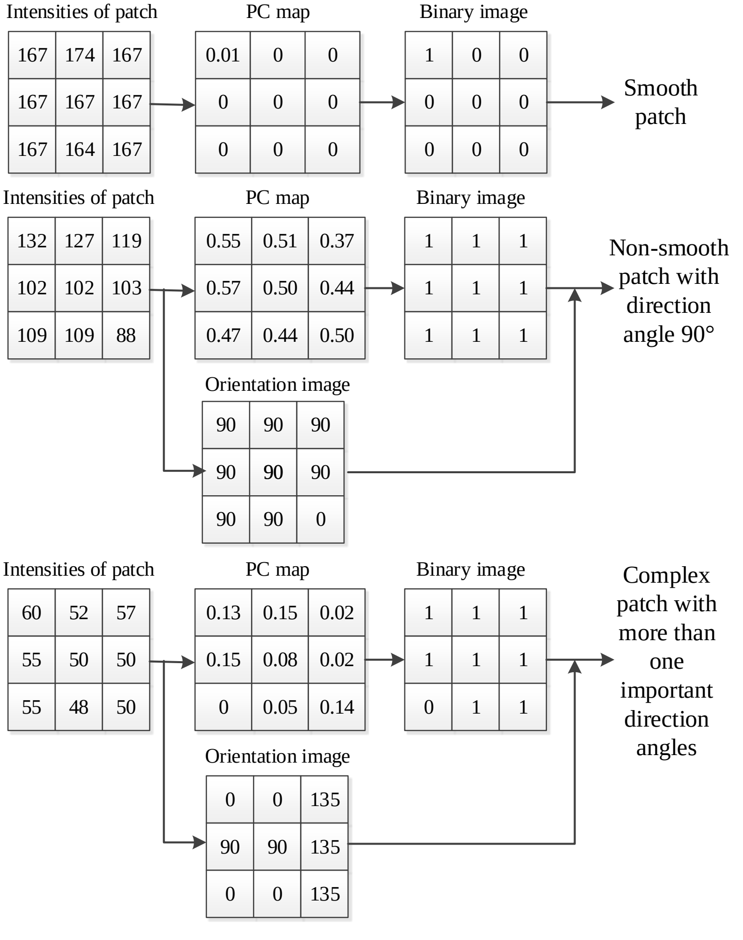


Figure 1. Examples of image patch classification

Extract patches from the binary image and orientation image, denote as $\{b_i\}$ and $\{o_i\}$. Count the number of '1' for each b_i, and if it is smaller than 1/3 of the total number of pixels in the patch, then patch i is classified as a smooth patch. For each non-smooth patch, find the direction angle d_i that repeats most times in o_i. Classify non-smooth patches that have identical d_i into the same category. However, if the number of occurrences of d_i is less than half of the total number of image patches, the pixel block may have more than one main orientation, and it is judged as a complex patch. So if we compute PC map for $J-1$ orientations, we will get J categories of patches. Figure 1 shows examples of image patch classification.

Using the above classification method, example patches are classified into different categories with similar patterns, and for each category a sub-dictionary can be learned. For an image patch to be processed, the sub-dictionary that belongs to the same category is selected adaptively.

2.3 SR Reconstruction Based on Sparse Coding with Multi-Class Dictionary (SC-MCD)

The algorithm consists of two parts: multi-class dictionaries training and SR reconstruction.

Part I: Multi-Class Dictionary Pairs Training (Can Be Done Offline)

The first step is to extract feature vector pairs from the HR and LR training images. Firstly, LR patches of size $n \times n$ pixels are extracted from the LR training images, and the classification algorithm described in Section 2.2 is applied to the patches, so each patch gets a category label $j, j = 1, 2, \cdots, J$. Secondly, high-pass filters are used to extract features from the LR training images. The four filters used are:

$$f_1 = [-1, 1], f_2 = f_1^T, f_3 = [1, -2, 1]/2, f_4 = f_3^T \tag{4}$$

where the superscript "T" means transpose. Applying these filters yields four vectors for each LR patch, which are concatenated into one vector q_l^j as the feature vector of the LR patch, where the superscript j is the category label of the patch. The feature vectors with the same category labels are grouped together, so the LR feature vector set $\left\{ \boldsymbol{Q}_l^{(1)}, \boldsymbol{Q}_l^{(2)}, \ldots, \boldsymbol{Q}_l^{(J)} \right\}$ is obtained. Finally, the HR training images are subtracted by the interpolated images scaled up from the corresponding LR training ones and the high frequency parts are kept. HR patches of size $R_n \times R_n$ pixels are extracted from the high frequency images, where R is the SR ratio. Each HR patch is arranged in vector form q_h^j as the feature vector, where the superscript j is the category label of the corresponding LR patch. The feature vectors with the same category labels are grouped together, so the HR feature vector set $\left\{ \boldsymbol{Q}_h^{(1)}, \boldsymbol{Q}_h^{(2)}, \ldots, \boldsymbol{Q}_h^{(J)} \right\}$ is obtained.

Suppose that there are totally J categories of feature vector pairs set for HR and LR image patches respectively, $\left\{ \left\langle \boldsymbol{Q}_l^{(1)}, \boldsymbol{Q}_h^{(1)} \right\rangle, \left\langle \boldsymbol{Q}_l^{(2)}, \boldsymbol{Q}_h^{(2)} \right\rangle, \ldots, \left\langle \boldsymbol{Q}_l^{(J)}, \boldsymbol{Q}_h^{(J)} \right\rangle \right\}$ for HR and LR image patches, respectively, with the same number of columns for each $\left\langle \boldsymbol{Q}_l^{(j)}, \boldsymbol{Q}_h^{(j)} \right\rangle$ pairs. We want to learn J LR sub-dictionaries $\left\{ \boldsymbol{D}_l^{(1)}, \boldsymbol{D}_l^{(2)}, \ldots, \boldsymbol{D}_l^{(J)} \right\}$

under which the patches in $\left\{\boldsymbol{Q}_l^{(1)}, \boldsymbol{Q}_l^{(2)}, \ldots, \boldsymbol{Q}_l^{(J)}\right\}$ can be sparsely represented, respectively. The problem can be written as:

$$\left\{\boldsymbol{D}_l^{(j)}, \boldsymbol{A}^{(j)}\right\} = \arg\min \left\|\boldsymbol{D}_l^{(j)}\boldsymbol{A}^{(j)} - \boldsymbol{Q}_l^{(j)}\right\|_2^2 \ s.t. \ \left\|\alpha_i^{(j)}\right\|_0 \leq C, j = 1, 2, \ldots, J \quad (5)$$

where $\boldsymbol{A}^{(j)}$ is the coding coefficients of the patch vectors set $\boldsymbol{Q}_l^{(j)}$ under the sub-dictionary $\boldsymbol{D}_l^{(j)}$, $\alpha_i^{(j)}$ is the i^{th} column of $\boldsymbol{A}^{(j)}$, $\left\|\alpha_i^{(j)}\right\|_0$ represents the number of non-zero coefficients in $\alpha_i^{(j)}$, C is the sparsity threshold. K-SVD dictionary learning algorithm [28] is used to simultaneously determine the sub-dictionaries and the coding coefficients.

The HR sub-dictionaries can be computed by the following Pseudo-Inverse expression:

$$\boldsymbol{D}_h^{(j)} = \boldsymbol{Q}_h^{(j)}\left(\boldsymbol{A}^{(j)}\right)^+ = \boldsymbol{Q}_h^{(j)}\boldsymbol{A}_h^{(j)T}\left(\boldsymbol{A}^{(j)}\boldsymbol{A}^{(j)T}\right)^{-1} \quad (6)$$

where Q_h and Q_l are matrices of feature vectors of HR and LR image patches, D_h and D_l are the HR and LR dictionaries, $Q_h = A_h D_h$, $Q_l = A_l D_l$, where A_h and A_l are sparse matrices. We suppose the two dictionaries D_h and D_l are trained to have the same sparse representations for each HR and LR image patch pair, so $A_h = A_l = A$. The coefficient matrix A is obtained during the training process of D_l, and the dictionary D_h is obtained from D_h and A from the formula

$$\boldsymbol{D}_h^{(j)} = \arg\min \left\|\boldsymbol{Q}_h^{(j)} - \boldsymbol{A}^{(j)}\boldsymbol{D}_h^{(j)}\right\|_F^2. \quad (7)$$

Multi-class dictionary pairs are trained, instead of only one pair of dictionary $\langle D_l, D_h\rangle$:

$$\left\{\left\langle\boldsymbol{D}_l^{(1)}, \boldsymbol{D}_h^{(1)}\right\rangle, \left\langle\boldsymbol{D}_l^{(2)}, \boldsymbol{D}_h^{(2)}\right\rangle, \ldots, \left\langle\boldsymbol{D}_l^{(J)}, \boldsymbol{D}_h^{(J)}\right\rangle\right\}. \quad (8)$$

Part II: SR Reconstruction

For a given LR test image $\boldsymbol{Y}$, patches of size $n \times n$ pixels are extracted, with $n - 1$ pixels overlap in each direction. For each patch, extract feature vector y, and calculate which category it belongs to. For each y that belongs to j-th category, sub-dictionary pair $\left\langle\boldsymbol{D}_l^{(j)}, \boldsymbol{D}_h^{(j)}\right\rangle$ is selected, and the sparse coding can be written as

$$\mu^{(j)*} = \arg\min \left\|\boldsymbol{D}_l^{(j)}\mu^{(j)} - y\right\|_2^2 \ s.t. \ \left\|\mu^{(j)}\right\|_0 \leq C. \quad (9)$$

OMP algorithm [29] is used to calculate the optimal solution $\mu^{(j)*}$, and the corresponding HR feature vector is obtained by $x = \boldsymbol{D}_h^{(j)}\mu^{(j)*}$. When all the HR features are obtained, the high frequency HR image $\boldsymbol{X}_k$ can be constructed by enforcing local compatibility and smoothness constraints between adjacent patches. The target HR image $\boldsymbol{X}_0$ is the summation of the high frequency image $\boldsymbol{X}_k$ and the interpolated image $\boldsymbol{X}_i$, which is scaled up from the LR image $\boldsymbol{Y}$ by bicubic interpolation.

The HR image produced by the above sparse representation method may not satisfy the reconstruction constraint exactly. IBP [27] is employed to alleviate this problem:

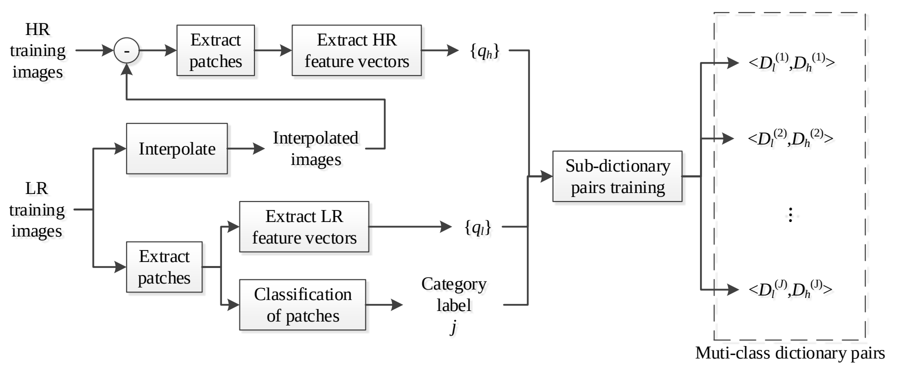
$$\boldsymbol{X}^* = \arg\min \|SH\boldsymbol{X} - \boldsymbol{Y}\|_2^2 . \tag{10}$$

The solution can be efficiently computed using gradient descent:

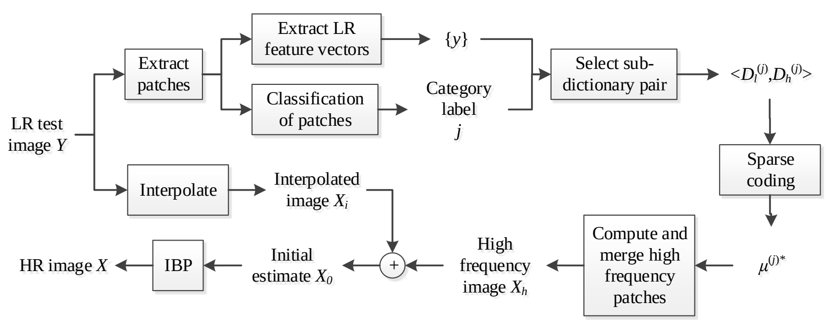
$$\boldsymbol{X}_{t+1} = \boldsymbol{X}_t + \tau \left[H^T S^T (\boldsymbol{Y} - SH\boldsymbol{X}_t) \right] . \tag{11}$$

where τ is the step size of the gradient descent, $\boldsymbol{X}_t$ is the estimate of HR image after the t^{th} iteration, and $\boldsymbol{X}_0$ is used as the initial estimate of $\boldsymbol{X}$.

The block diagram of the proposed SC-MCD algorithm is shown in Figure 2.



a) Multi-class dictionary pairs training



b) SR reconstruction

Figure 2. The block diagram of the proposed SC-MCD algorithm

3 EXPERIMENTAL RESULTS

We use several benchmarks, including Set14 [14], Set5 [30] and B100 [16] as our testing set. The magnification factor is 4. Those 3×3 patches are extracted in LR images, and the corresponding HR patches are 12×12. About 80 000 training patch pairs are collected from the training image set used in [13] and 1 024 atoms are trained for each sub-dictionary. Sparsity threshold C is set to be 5. All the simulations are conducted in MATLAB R2016a on PC with Intel® Core™ i7/3.6 GHz/4 GB.

If the number of categories is too small, the differences between the geometric features of the image patches corresponding to each sub-dictionary will be rather large. On the other hand, if there are a lot of categories, the discriminations between sub-dictionaries are low. Therefore, both the factors of accuracy and discrimination are taken into account in the expression of sub-dictionaries to choose the appropriate number of categories. In the experiments, the number of categories is increased from 5 to 9, and the average peak signal-noise ratio (PSNR) of each data set is calculated. The results are shown in Table 1. Experiments show that the reconstruction results are better when the number of categories (J) is set to 6 or 7. In the later experiments, J is set to be 6.

	J = 5	J = 6	J = 7	J = 8	J = 9
Set5	27.787	27.787	**27.814**	27.785	27.762
Set14	25.095	**25.117**	25.101	25.095	25.097
B100	25.331	**25.335**	25.332	25.327	25.324

Table 1. Mean PSNR of SR reconstructed images using different number of categories

3.1 MCD vs. SCD

In this part, we evaluate the influence of multi-class dictionaries to the quality of reconstructed HR images. We test two methods: SR reconstruction with single class dictionary (denote as SCD) [14], and SR reconstruction with multi-class dictionaries (denote as MCD). In order to find out the contribution of the multi-class dictionaries independently, we do not apply IBP to enforce global reconstruction constraint in the experiments for this part.

We select 7 images from Set14 as the testing set. For each test image, the mean squared error (MSE) between the reconstructed image and the original one is calculated, and the result is shown in Table 2. We can see that, compared to SCD, MCD averagely reduces the squared error by 7.541 per pixel.

3.2 MD_PC vs. MD_KM

In this part, we compare the SR performance by different multi-class dictionaries. The proposed image patches classification method based on phase congruency is

	SCD	MCD
Barbara	271.231	253.994
Coastguard	237.873	233.966
Face	77.549	76.640
Foreman	109.968	99.542
Man	200.533	197.136
Pepper	108.982	105.698
Zebra	252.086	238.458
Average	179.746	172.205

Table 2. MSE of SR reconstructed images using SCD and MCD

denoted as MD_PC. In [25] Dong et al. use k-means for image patches clutering, denoted as MD_KM. In order to highlight the impact of the image patches classification algorithms on SR performance, the SR reconstruction process here does not introduce any global reconstruction constraints.

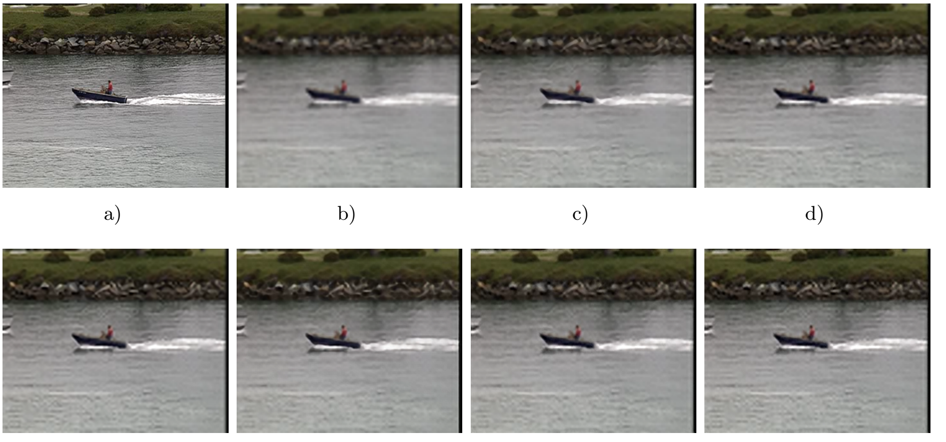
Measures	MD_KM		MD_PC	
	PSNR (dB)	FSIM	PSNR (dB)	FSIM
Barbara	24.038	0.933	24.083	0.938
Coastgrard	24.337	0.725	24.439	0.729
Face	29.322	0.871	29.286	0.870
Foreman	27.060	0.879	28.151	0.896
Man	25.098	0.932	25.183	0.936
Pepper	27.713	0.961	27.890	0.965
Zebra	24.075	0.909	24.357	0.920
Average	25.949	0.887	26.198	0.894

Table 3. Numerical measurements of the reconstructed images by multi-class dictionaries

The results are compared by PSNR and Feature Similarity (FSIM) [31]. The higher of PSNR and FSIM means much similar of the reconstructed image to the original image. As shown in Table 3, the numerical measurements of PSNR and FSIM obtained by proposed MD_PC are higher than that of MD_KM used in [25].

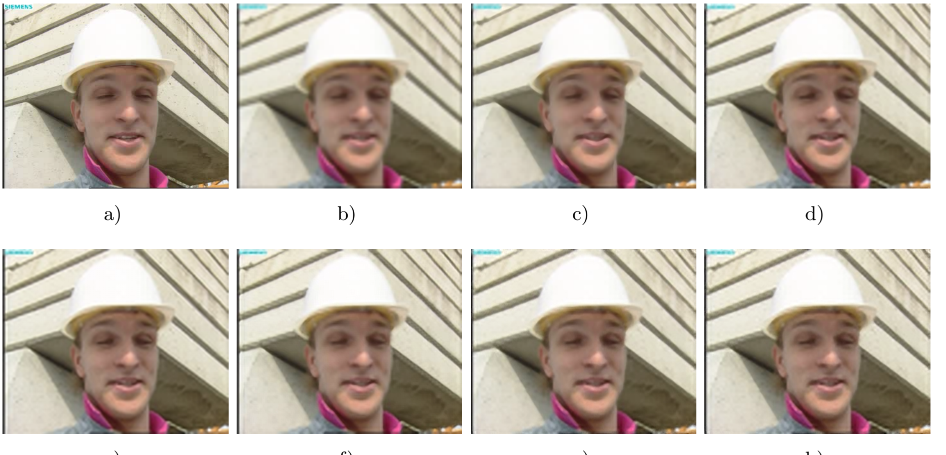
3.3 SC_MCD vs. Other Algorithms

We compare the proposed MCD (without IBP) and MCD_IBP methods with Bicubic interpolation method, Kim's method using sparse regression and natural image prior denoted as SR_NIP [32], Zeyde's sparse coding-based SR using a universal dictionary denoted as SC_SR [14], Dong's method by k-means clustering, adaptive sparse domain selection and adaptive regularization denoted as ASDS_AR_NL [25], and Dong's method by deep convolutional neural network denoted as SRCNN [18]. The comparison experiments are based on the matlab versions of the source code



a) b) c) d) e) f) g) h)

Figure 3. Original "Coastguard" image and images reconstructed by different methods. a) Original image, b) Bicubic, c) SR_NIP [32], d) SC_SR [14] e) ASDS_AR_NL [25], f) SR-CNN [18], g) MCD, h) MCD_IBP.



a) b) c) d) e) f) g) h)

Figure 4. Original "Foreman" image and images reconstructed by different methods. a) Original image, b) Bicubic, c) SR_NIP [32], d) SC_SR [14], e) ASDS_AR_NL [25], f) SRCNN [18], g) MCD, h) MCD_IBP.

or demos provided by the above papers, so it is very fair. The results are compared by visual quality subjectively and by numerical measurements of PSNR and FSIM.

Image	Measures	Bicubic	SR_NIP [32]	SC_SR [14]	ASDS_AR_NL [25]	SRCNN [18]	MCD	MCD_IBP
Baboon	PSNR (dB)	20.199	20.436	20.407	20.476	20.423	20.476	**20.499**
	FSIM	0.849	0.890	0.892	**0.904**	0.886	**0.904**	**0.904**
Barbara	PSNR (dB)	23.440	24.077	23.797	**24.120**	23.982	24.083	24.118
	FSIM	0.896	0.934	0.931	0.936	0.932	**0.938**	**0.938**
Bridge	PSNR (dB)	22.873	23.440	23.519	23.561	23.421	23.552	**23.610**
	FSIM	0.870	0.904	0.909	0.913	0.901	**0.916**	**0.916**
Coastguard	PSNR (dB)	23.821	24.333	24.367	24.374	24.339	24.439	**24.479**
	FSIM	0.639	0.685	0.696	0.724	0.692	**0.729**	**0.729**
Comic	PSNR (dB)	22.022	20.767	20.730	**20.909**	20.820	20.782	20.841
	FSIM	0.702	0.750	0.749	**0.765**	0.760	0.761	0.761
Face	PSNR (dB)	28.620	29.156	29.235	**29.437**	29.015	29.286	29.333
	FSIM	0.826	0.845	0.856	**0.872**	0.850	0.870	0.870
Flowers	PSNR (dB)	23.560	24.552	24.403	**24.662**	24.601	24.491	24.548
	FSIM	0.770	0.809	0.806	**0.822**	0.815	0.819	0.819
Foreman	PSNR (dB)	25.756	27.640	27.718	27.553	26.596	28.151	**28.222**
	FSIM	0.850	0.888	0.888	0.890	0.882	**0.896**	**0.896**
Lena	PSNR (dB)	27.973	29.186	29.040	**29.333**	29.029	29.151	29.201
	FSIM	0.935	0.959	0.960	**0.967**	0.956	0.965	0.965
Man	PSNR (dB)	24.187	25.234	25.109	**25.315**	25.212	25.183	25.251
	FSIM	0.892	0.930	0.930	**0.939**	0.929	0.936	0.936
Monarch	PSNR (dB)	25.867	27.780	27.300	27.764	**28.125**	27.419	27.517
	FSIM	0.922	0.954	0.945	**0.956**	**0.956**	0.949	0.949
Pepper	PSNR (dB)	26.974	27.864	27.757	**27.923**	27.660	27.890	27.918
	FSIM	0.937	0.963	0.962	**0.967**	0.960	0.965	0.965
PPT3	PSNR (dB)	20.203	21.465	21.341	21.435	**21.920**	21.569	21.635
	FSIM	0.824	0.883	0.886	0.892	**0.896**	0.889	0.889
Zebra	PSNR (dB)	22.438	24.380	24.115	24.329	24.457	24.357	**24.466**
	FSIM	0.851	0.914	0.910	0.919	0.916	**0.920**	**0.920**
Average	PSNR (dB)	23.995	25.022	24.917	25.085	24.971	25.059	**25.117**
	FSIM	0.840	0.879	0.880	**0.890**	0.881	**0.890**	**0.890**

Table 4. Numerical measurements of the reconstructed images of Set14 by different methods

Figure 3 shows the original "Coastguard" image and images reconstructed by different methods. Figure 4 gives the results of "Foreman" image. Table 4 shows numerical measurements of the reconstructed images of Set14. Table 5 shows the average PSNR and FSIM measurements on several benchmarks, including Set5, Set14 and B100. We can see ASDS_AR_NL [25], the proposed MCD and MCD_IBP methods have best results, which outperform SR_NIP [32], SC_SR [14] and SRCNN [18].

Method	Measures	Set5	Set14	B100
Bicubic	PSNR (dB)	26.226	23.995	24.463
	FSIM	0.837	0.840	0.791
SR_NIP [32]	PSNR (dB)	27.638	25.022	25.220
	FSIM	0.879	0.879	0.784
SC_SR [14]	PSNR (dB)	27.569	24.917	25.204
	FSIM	0.872	0.880	0.793
ASDS_AR_NL [25]	PSNR (dB)	**27.951**	25.085	**25.355**
	FSIM	0.883	**0.890**	**0.812**
SRCNN [18]	PSNR (dB)	27.677	24.971	25.200
	FSIM	**0.885**	0.881	0.791
MCD_IBP	PSNR (dB)	27.787	**25.117**	25.332
	FSIM	0.878	**0.890**	**0.812**

Table 5. Average numerical measurements of the reconstructed images of Set 5, Set14 and B100

Image	SR_NIP [32]	SC_SR [14]	ASDS_AR_NL [25]	SRCNN [18]	MCD	MCD_IBP
Baboon	31.596	1.583	103.039	10.481	1.897	1.954
Barbara	44.494	2.679	185.560	19.267	3.306	3.391
Bridge	41.816	1.684	117.051	11.446	2.061	2.127
Coastguard	11.041	0.635	36.238	2.574	0.775	0.817
Comic	21.218	0.557	34.803	2.410	0.683	0.706
Face	4.294	0.471	25.832	2.047	0.575	0.603
Flowers	25.238	1.163	75.181	7.510	1.398	1.452
Foreman	13.245	0.635	36.788	2.671	0.776	0.803
Lenna	23.240	1.684	110.869	11.246	2.048	2.138
Man	39.288	1.682	112.917	11.226	2.070	2.135
Monarch	39.786	2.521	183.439	17.681	3.099	3.207
Pepper	24.084	1.669	110.913	11.209	2.067	2.127
PPT3	46.084	2.102	163.759	16.059	2.303	2.392
Zebra	39.680	1.436	99.953	9.544	1.764	1.841
Average	28.936	**1.464**	99.739	9.669	**1.773**	**1.835**

Table 6. Reconstruction time(s) of different methods

Table 6 shows the reconstruction time to investigate the time complexity. Among the four algorithms with the best reconstruction effect, the proposed MCD and MCD_IBP are much faster than SRCNN and ASDS_AR_NL. The average time for reconstructing an image is less than 2 seconds using the proposed MCD and MCD_IBP algorithms, whereas the SRCNN needs about 10 seconds and ASDS_AR_NL needs nearly 100 seconds.

Considering all the factors including subjective visual quality, objective assessment and time complexity, the proposed method obtains good performance for image

SR reconstruction. The proposed methods not only significantly improve the image reconstruction speed, but also significantly improve the image reconstruction quality.

4 CONCLUSIONS

In this paper, we propose a SR reconstruction algorithm based on sparse coding with multi-class dictionaries. A novel method for image patch classification is put forward, and a sub-dictionary is selected adaptively for each given image patch. IBP is used to enforce global reconstruction constraint. Our approach produces comparable or even better SR reconstruction results with some existing algorithms. The robustness of the proposed algorithm under different imaging conditions will be our future work.

Acknowledgment

This work is supported by Young Creative Talents Project of Department of Education of Guangdong Province (Natural Science) (2016KQNCX092, 2017KQNCX117, 2015KQNCX084, 2016KQNCX089), National Natural Science Foundation of China (61772144, 61702119), Innovation Research Project of Education Department of Guangdong Province (Natural Science) (2016KTSCX077), Foreign Science and Technology Cooperation Plan Project of Guangzhou Science Technology and Innovation Commission (201807010059), National Natural Science Foundation of Guangdong (2016A030313472, 2018A030313994, 2018A0303130187), Science and Technology Program of Guangzhou (201607010152).

REFERENCES

[1] Park, S. C.—Park, M. K.—Kang, M. G.: Super-Resolution Image Reconstruction: A Technical Overview. IEEE Signal Processing Magazine, Vol. 20, 2003, No. 3, pp. 21–36, doi: 10.1109/MSP.2003.1203207.

[2] Zhang, K.—Zuo, W.—Zhang, L.: Learning a Single Convolutional Super-Resolution Network for Multiple Degradations. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 3262–3271, doi: 10.1109/CVPR.2018.00344.

[3] Pandey, G.—Ghanekar, U.: A Compendious Study of Super-Resolution Techniques by Single Image. Optik, Vol. 166, 2018, pp. 147–160, doi: 10.1016/j.ijleo.2018.03.103.

[4] Freeman, W. T.—Pasztor, E. C.—Carmichael, O. T.: Learning Low-Level Vision. International Journal of Computer Vision (IJCV), Vol. 40, 2000, No. 1, pp. 25–47, doi: 10.1023/A:1026501619075.

[5] Stephenson, T. A.—Chen, T.: Adaptive Markov Random Fields for Example-Based Super-Resolution of Faces. EURASIP Journal on Advances in Signal Processing, 2006, Art. No. 031062, 11 pp., doi: 10.1155/ASP/2006/31062.

[6] Ma, Y. J.—Zhang, H.—Xue, Y.—Zhang, S.: Super-Resolution Image Reconstruction Based on K-Means-Markov Network. Proceedings of the 2009 Fifth International Conference on Natural Computation, Tianjin, China, IEEE, 2009, pp. 316–318, doi: 10.1109/ICNC.2009.608.

[7] Chang, H.—Yeung, D.-Y.—Xiong, Y.: Super-Resolution Through Neighbor Embedding. Proceedings of the 2004 IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2004), 2004, pp. 275–282, doi: 10.1109/CVPR.2004.1315043.

[8] Zhang, Z.—Qi, C.—Hao, Y.: Locality Preserving Partial Least Squares for Neighbor Embedding-Based Face Hallucination. Proceedings of the 2016 IEEE International Conference on Image Processing (ICIP), 2016, pp. 409–413, doi: 10.1109/ICIP.2016.7532389.

[9] Rahiman, V. A.—George, S. N.: Single Image Super Resolution Using Neighbor Embedding and Statistical Prediction Model. Computer and Electrical Engineering, Vol. 62, 2017, pp. 281–292, doi: 10.1016/j.compeleceng.2016.12.018.

[10] Gao, X.—Zhang, K.—Tao, D.—Li, X.: Image Super-Resolution with Sparse Neighbor Embedding. IEEE Transactions on Image Processing, Vol. 21, 2012, No. 7, pp. 3194–3205, doi: 10.1109/TIP.2012.2190080.

[11] Wang, S.—Zhang, L.—Liang, Y.—Pan, Q.: Semi-Coupled Dictionary Learning with Applications to Image Super-Resolution and Photo-Sketch Synthesis. Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition, 2012, pp. 2216–2223, doi: 10.1109/CVPR.2012.6247930.

[12] He, L.—Qi, H.—Zaretzki, R.: Beta Process Joint Dictionary Learning for Coupled Feature Spaces with Application to Single Image Super-Resolution. Proceedings of the 2013 IEEE Conference on Computer Vision and Pattern Recognition, 2013, pp. 345–352, doi: 10.1109/CVPR.2013.51.

[13] Yang, J.—Wright, J.—Huang, T. S.—Ma, Y.: Image Super-Resolution via Sparse Representation. IEEE Transactions on Image Processing, Vol. 19, 2010, No. 11, pp. 2861–2873, doi: 10.1109/TIP.2010.2050625.

[14] Zeyde, R.—Elad, M.—Protter, M.: On Single Image Scale-Up Using Sparse-Representations. Proceedings of the 7th International Conference on Curves and Surfaces, 2010. Springer, Berlin, Heidelberg, Lecture Notes in Computer Science, Vol. 6920, 2012, pp. 711–730, doi: 10.1007/978-3-642-27413-8_47.

[15] Timofte, R.—De, V.—Van Gool, L.: Anchored Neighborhood Regression for Fast Example-Based Super-Resolution. Proceedings of the 2013 IEEE International Conference Computer Vision, 2013, pp. 1920–1927, doi: 10.1109/ICCV.2013.241.

[16] Timofte, R.—De Smet, V.—Van Gool, L.: A+: Adjusted Anchored Neighborhood Regression for Fast Super-Resolution. In: Cremers, D., Reid, I., Saito, H., Yang, M. H. (Eds.): Computer Vision – ACCV 2014. Springer, Cham, Lecture Notes in Computer Science, Vol. 9006, 2014, pp. 111–126, doi: 10.1007/978-3-319-16817-3_8.

[17] Schulter, S.—Leistner, C.—Bischof, H.: Fast and Accurate Image Upscaling with Super-Resolution Forests. Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 3791–3799, doi: 10.1109/CVPR.2015.7299003.

[18] Dong, C.—Loy, C. C.—He, K.—Tang, X.: Image Super-Resolution Using Deep Convolutional Networks. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 38, 2016, No. 2, pp. 295–307, doi: 10.1109/TPAMI.2015.2439281.

[19] Kim, J.—Lee, J. K.—Lee, K. M.: Accurate Image Super-Resolution Using Very Deep Convolutional Networks. Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 1646–1654, doi: 10.1109/CVPR.2016.182.

[20] Lai, W.-S.—Huang, J.-B.—Ahuja, N.—Yang, M.-H.: Deep Laplacian Pyramid Networks for Fast and Accurate Super-Resolution. Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 5835–5843, doi: 10.1109/CVPR.2017.618.

[21] Liu, D.—Wang, Z.—Wen, B.—Yang, J.—Han, W.—Huang, T. S.: Robust Single Image Super-Resolution via Deep Networks with Sparse Prior. IEEE Transactions on Image Processing, Vol. 25, 2016, No. 7, pp. 3194–3207, doi: 10.1109/TIP.2016.2564643.

[22] Yang, C.-Y.—Yang, M.-H.: Fast Direct Super-Resolution by Simple Functions. Proceedings of the 2013 IEEE International Conference on Computer Vision, 2013, pp. 561–568, doi: 10.1109/ICCV.2013.75.

[23] Elad, M.—Yavneh, I.: A Plurality of Sparse Representation Is Better Than the Sparsest One Alone. IEEE Transactions on Information Theory, Vol. 55, 2009, No. 10, pp. 4701–4714, doi: 10.1109/TIT.2009.2027565.

[24] Yang, S.—Liu, Z.—Wang, M.—Sun, F.—Jiao, L.: Multitask Dictionary Learning and Sparse Representation Based Single-Image Super-Resolution Reconstruction. Neurocomputing, Vol. 74, 2011, No. 17, pp. 3193–3203, doi: 10.1016/j.neucom.2011.04.014.

[25] Dong, W.—Zhang, L.—Shi, G.—Wu, X.: Image Deblurring and Super-Resolution by Adaptive Sparse Domain Selection and Adaptive Regularization. IEEE Transactions on Image Processing, Vol. 20, 2011, No. 7, pp. 1838–1857, doi: 10.1109/TIP.2011.2108306.

[26] Kovesi, P.: Phase Congruency: A Low-Level Image Invariant. Psychological Research, Vol. 64, 2000, No. 2, pp. 136–148, doi: 10.1007/s004260000024.

[27] Irani, M.—Peleg, S.: Improving Resolution by Image Registration. CVGIP: Graphical Models and Image Processing, Vol. 53, 1991, No. 3, pp. 231–239, doi: 10.1016/1049-9652(91)90045-L.

[28] Aharon, M.—Elad, M.—Bruckstein, A.: K-SVD: An Algorithm for Designing Overcomplete Dictionaries for Sparse Representation. IEEE Transactions on Signal Processing, Vol. 54, 2006, No. 11, pp. 4311–4322, doi: 10.1109/TSP.2006.881199.

[29] Pati, Y. C.—Rezaiifar, R.—Krishnaprasad, P. S.: Orthogonal Matching Pursuit: Recursive Function Approximation with Applications to Wavelet Decomposition. Proceedings of the 27th Asilomar Conference on Signals, Systems and Computers, 1993, pp. 40–44, doi: 10.1109/ACSSC.1993.342465.

[30] Bevilacqua, M.—Roumy, A.—Guillemot, C.—Alberi Morel, M.-L.: Low-Complexity Single-Image Super-Resolution Based on Nonnegative Neighbor Embedding. Proceedings of the British Machine Vision Conference, 2012, pp. 135.1–135.10, doi: 10.5244/C.26.135.

[31] Zhang, L.—Zhang, L.—Mou, X.—Zhang, D.: FSIM: A Feature Similarity Index for Image Quality Assessment. IEEE Transactions on Image Processing, Vol. 20, 2011, No. 8, pp. 2378–2386, doi: 10.1109/TIP.2011.2109730.

[32] Kim, K. I.—Kwon, Y.: Single-Image Super-Resolution Using Sparse Regression and Natural Image Prior. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 32, 2010, No. 6, pp. 1127–1133, doi: 10.1109/TPAMI.2010.25.

Xiuxiu Liao received her Ph.D. degree in computer application technology from the South China University of Technology in 2013. Her research interests include image processing, compressive sensing and machine learning. She is currently working as Lecturer in the School of Computer Science, Guangdong Polytechnic Normal University, Guangzhou, China.

Kejia Bai received his Ph.D. in system engineering from the South China University of Technology, Guangzhou, China, in 2009. He is currently Associate Professor in the Guangdong Polytechnic Normal University, Guangzhou, China. His research interests include image processing, pattern recognition, artificial intelligence, super-resolution, and target tracking.

Qian Zhang received her Ph.D. degree in computer application technology from the South China University of Technology in 2013. Her research interests include image processing and deep learning. She is currently working as Lecturer in the School of Computer Science, Guangdong Polytechnic Normal University, Guangzhou, China.

Xiping Jia received his Ph.D. degree in computer application technology from the South China University of Technology in 2008. His research interests include image processing, data mining and machine learning. He is currently working as Associate Professor in the School of Computer Science, Guangdong Polytechnic Normal University, Guangzhou, China.

Shaopeng Liu received his Ph.D. degree from the Sun Yat-Sen University, Guangdong, China, in 2013. Currently, he is Lecturer in the School of Computer Science, Guangdong Polytechnic Normal University, China. His research interests include machine learning, computational intelligence and medical image analysis.

Jin Zhan received her B.Sc. and Ph.D. degrees in computer application technology from the Sun Yat-Sen University in 2004 and 2015, respectively, and she is currently Associate Professor in the School of Computer Sciences, Guangdong Polytechnic Normal University. Her research interests include image processing, video analysis, machine learning and applications.

Computing and Informatics, Vol. 38, 2019, 1320–1340, doi: 10.31577/cai_2019_5_1320

PARALLEL PEER GROUP FILTER FOR IMPULSE DENOISING IN DIGITAL IMAGES ON GPU

José Agustín Tortolero Osuna, Alberto Jorge Rosales Silva

Instituto Politécnico Nacional de México, ESIME Zacatenco
Av. IPN s/n, Edif. Z, Acceso 3, 3er Piso, SEPI-Electrónica
Col. Lindavista, 07738, CDMX, México
e-mail: arosaless@ipn.mx

Abstract. A new two-steps impulsive noise parallel Peer Group filter for color images using Compute Unified Device Architecture (CUDA) on a graphic card is proposed. It consists of two steps: impulsive noise detection, which uses a Fuzzy Metric as a distance criterion and a filtering step. For the needed ordering algorithm we are using the Marginal Median Filter with forgetful selection sort. Comparisons with other color filters for Graphics Processing Unit (GPU) architectures are presented, demonstrating that our proposal presents better performance in color preservation and noise suppression.

Keywords: Impulse denoising, color images, CUDA, parallel architecture, fuzzy metrics

1 INTRODUCTION

The growing demand for digital image processing algorithms for applications, i.e. in consumer electronics, industry, and medicine, creates the need to develop algorithms to be able to perform itself quickly and executed accordingly to the needs of each application field. All processing algorithms must possess, either hardware or software, denoising procedures to improve the image information obtained from the acquisition hardware device, by doing this the information provided is more significant for other post-processing stages (i.e. face recognition, augmented reality, computer vision, etc.). Noise affects the majority of image processing systems, being responsible for the 99 % of the failures [1, 2, 3, 4]. This clearly justifies the use of noise-suppressing algorithms incorporated within the systems. The incorporation of a denoising algorithm as a pre-processing stage consumes hardware and software resources which take up processing time affecting other post-processing procedures.

Decades ago most of the designed filters were carried out serially, today the current processing needs lead to optimize resources and make processing times more efficient [5]. This points to parallel implementations; this type of processing leads to substantially improving computational time in all stages of an image processing system. In this way, it is proposed to exploit the potentials of parallelism and use them for noise suppression. Between the noise types that affect the digital images, the impulsive noise is one of the most important, this affects digital images in the acquisition or transmission stages and through malfunctioning sensors or communication channels [2, 3, 4], this is produced by human-made phenomena, such as car ignition systems, industrial machines in the vicinity of the receiver, switching transients in power lines, unprotected switches, natural causes, etc. Impulsive noise has been treated using parallel architectures, e.g. the Peer Group Family filters [6], presenting acceptable results preserving inherent characteristics of the digital images. Their main features are that they are fast and have low computational complexity, these characteristics are used in parallel systems for impulse noise removal.

So, our efforts are focused on the design of a more robust and fast parallel filter for GPU architectures filter, that preserves inherent characteristics of the objects such as edges, details and chromatic content. We demonstrated the improved results against obtained from other states of the art algorithms reported such as the Peer Group Algorithm with Fuzzy Metric for GPU [6, 7, 8, 9].

Impulse noise in color images is modeled with random and fixed values, if $F^{\chi}_{(x,y)}$ denotes the input noisy image with χ denoting each color channel $\{R, G, B\}$, and $O^{\chi}_{(x,y)}$ denoting the original noise-free image for every (x, y) pixel position [2], this is described by Equation (1):

$$F^{\chi}_{(x,y)} = \begin{cases} O^{\chi}_{(x,y)}, & \text{with probability } 1-p, \\ \zeta^{\chi}_{(x,y)}, & \text{with probability } p, \end{cases} \tag{1}$$

where p is defined as the noise density for the random value of the impulsive noise, $\zeta^{\chi}_{(x,y)}$ is an identically distributed, independent random process with an arbitrary underlying probability density function, having values of impulsive noise in the interval $[0, \dots, 255]$, and for the fixed values (Salt and Pepper noise) the values present are denoted by 0 or 255. In Figure 1, the corrupted Peppers image with random and fixed noise values are shown.

The structure of the paper is as follows: in Section 2 a brief review of the GPU architecture is presented; in Section 3 the Peer Group Algorithm is introduced using fuzzy metric to justify the novelty of our proposal; Section 4 describes the proposed filtering algorithm in a detailed manner; in Section 5 a comparative analysis of the read/write memory accesses between the proposed and the state-of-the-art filters is done. Finally, Sections 5 and 6 show the performance results.

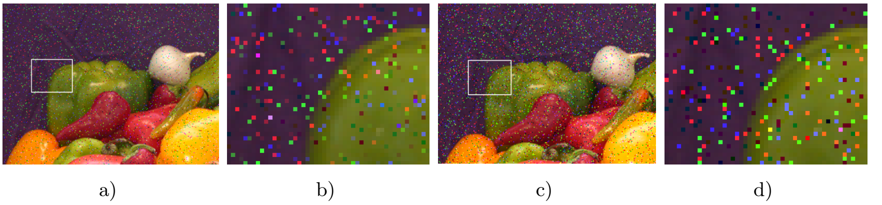


Figure 1. Peppers images with: a) random impulsive noise, c) salt and pepper impulsive noise, b) and d) the zoomed regions to denote impulsive distortions, respectively

2 GPU ACCELERATION CONSIDERATIONS FOR IMAGE PROCESSING

One of the main differences between CPU and GPU is the number of working threads. CPU can execute only up to two threads per core, while GPU chips can run thousands. Figure 2 shows an overview of the GPU card hardware consisting of Graphics Double Data Rate (GDDR) Memory (Global, Constant and Texture), Streaming Multiprocessors (SMs), and Streaming Processors (SPs) [10, 11].

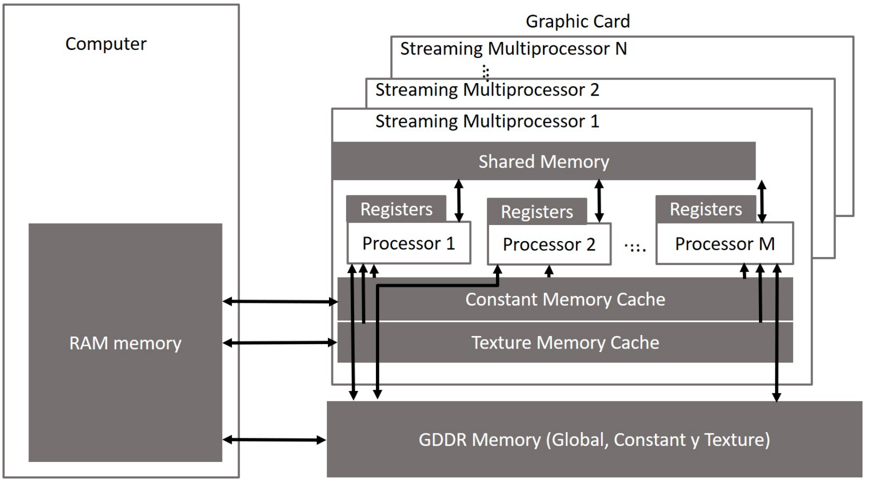


Figure 2. The GPU hardware [10]

The SPs are responsible for the mathematic operations done in the GPU chip. SMs are made of several Streaming Processors, up to 48 in modern GPUs; this is a crucial aspect for GPU scaling. A GPU has one or more SMs, where each has only one control unit; this unit seeks and decodes each instruction. The SMs on the GPU uses the Single Instruction Multiple Data (SIMD) architecture and the SP receives the same controlling signal and executes the same instruction set using different operands [12]. The optimum performance of parallel algorithms in graphic cards lies in the understanding of how memory devices work [12], what care should be taken in the type of memory where data is temporarily stored [13], because a bad memory selection could produce poor parallel performance, i.e., the off-chip memory offers more storing space and global scope but its writing and reading latency is high.

Another consideration is with respect to the registers which allow memory access without latency, unfortunately there are only 255 registers per thread used in the GPUs Maxwell architecture and shared memory (which is faster than off-chip memory). This memory is slower than register memory, it should be taken into consideration that bank conflicts can occur if two threads in the SM try to access data stored in the shared memory, in such cases, the parallel process is serialized, which may cause a significant performance decrease [12, 13]. For this reason, it is justifiable to take into account the selection of the correct memory to be used. The GPU graphic card has its own Graphics Double Data Rate (GDDR) memory where read/write Global, read-only Constant and Texture memories are allocated; the last two memories are used for speeding up the reading from the GDDR memory. The register memory is the fastest in the graphic card, but it is limited in quantity (i.e., on Fermi each SMs has 32 k memory addresses). Shared memory is the second-fastest memory and can be read by all the threads within the SM. The Global memory is for general purposes and is the slowest memory in the graphic card. These facts are summarized in Table 1.

Memory Type	Registers	Shared Memory	Texture Memory	Constant Memory	Global Memory
Bandwidth	Highest	High	Low	Low	Low
Location	On-chip	On-chip	Off-chip	Off-chip	Off-chip
Scope	Thread	Threads in SM	All SMs	All SMs	All SMs

Table 1. Graphic card memory overview

CUDA programming model is the tool to be used to exploit parallelism capabilities of the GPU card, this tool introduces three key abstractions [14]: a hierarchy of thread groups, shared memories, and barrier synchronization; these abstractions are presented to the programmer as a minimal set of language extensions. This decomposition preserves language expressivity by allowing threads to cooperate when solving each sub-problem, and at the same time enables automatic scalability. Each block of threads can be scheduled concurrently or sequentially on any of the available multiprocessors within a GPU, so a compiled CUDA program can execute on any number of multiprocessors, and only the runtime system needs to know the physical multiprocessor count. A CUDA-enabled program, called "kernel", can be executed in any number of SMs, as illustrated by Figure 3 [14].

Graphic cards can solve time-consuming image processing algorithms specifically in image denoising. Our proposal offers good quality results exploiting new tech-

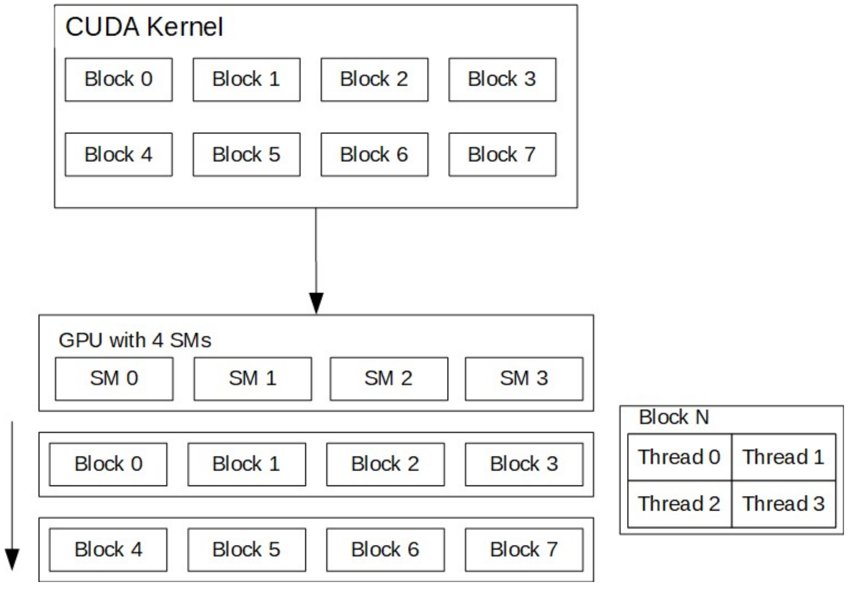


Figure 3. CUDA thread hierarchy

nologies capabilities and delivers optimum data pre-processed (data denoised) to other post-processing stages described before. Now in the next sections, we analyze, treat and discuss the proposal how the denoising algorithm is to be implemented in the GPU and its main characteristics to be taken into account to speed up the denoising stage.

3 PEER GROUP ALGORITHM

The Peer Group algorithm is a robust filter consisting in a processing window of $W = n \times n$ over a central pixel P_c^{χ}, the Peer Group $\mathcal{P}(P_c^{\chi}, \kappa, d)$ of such central pixel are the pixels satisfying $d(P_c^{\chi}, P_j^{\chi}) \leq \alpha$ where $d(P_c^{\chi}, P_j^{\chi})$ is an appropriate distance criterion between P_c^{χ} and one of its neighbors, P_j^{χ}, $j = n^2 - 1$, where n is the size of the processing window, with $n = 3$, and α is a fixed threshold [6]. The distance criterion used is the Euclidean distance described by Equation (2):

$$d(P_c^{\chi}, P_j^{\chi}) = \sqrt{\sum_{\chi=\{R,G,B\}} \left(P_c^{\chi} - P_j^{\chi}\right)^2}. \tag{2}$$

For every pixel within the processing window, if $d(P_c^{\chi}, P_j^{\chi}) \leq \alpha$ is fulfilled, then the pixel is labeled as a noise-free pixel, otherwise it is noisy. In the filtering stage, the noisy pixels are substituted by the substitute obtained from the Arithmetic Mean Filer (AMF) [7] or the Vector Median Filter (VMF) [15] applied to the set of free-noisy pixels. Sanchez et al. [7, 8, 9], propose the Peer Group algorithm with Fuzzy Metric (PGMF) for GPU architectures, where each stage is executed in separated kernels in serial sequence. This algorithm uses the Fuzzy Metric shown in Equation (3) [7, 8, 9, 16] instead of the Euclidean distance in the detection stage.

$$M_\infty = \prod_{\chi=1}^{3} \frac{\min(P_c^\chi, P_j^\chi) + K}{\max(P_c^\chi, P_j^\chi) + K}, \text{ with } K = 1\,024. \tag{3}$$

The kernel algorithm for detection and suppression of noise is shown in Algorithm 1 and Algorithm 2, respectively.

Algorithm 1 PGMF algorithm

function DetectionKernel ▷ Input: noisyImage ▷ Output: detectedNoise
    for each image pixel P_{th} its corresponding processing parallel thread th **do**
        $a_c^\chi \leftarrow \{R,G,B\}$ values of P_{th} from the global memory
        for all the P_j^χ pixels within W **do**
            $b_j^\chi \leftarrow \{R,G,B\}$ values of P_j^χ
            distance = Fuzzymetric(a_c^χ, b_j^χ)
            if distance $\geq \alpha$ **then**
                pixel $b_j^\chi \in \mathcal{P}(P_c^\chi, \kappa, \alpha)$
            end if
        end for
        if $\#\mathcal{P}(P_c^\chi, \kappa, \alpha) \geq (\kappa + 1)$ **then**
            P_{th} is labeled as noise free pixel
        else
            P_{th} is labeled as noisy pixel
        end if
    end for
end function

Algorithm 2 PGMF algorithm

function NoiseFilteringkernel ▷ Input: noisyImage ▷ Output: filteredImage
    for each image pixel P_{th} its corresponding processing thread th **do**
        for all the P_j^χ labeled as noise free pixels within W **do**
            FilteringArray$_j^\chi \leftarrow \{R,G,B\}$ values of P_j^χ
        end for
        Filter FilteringArray$_j^\chi$ use either the VMF or the AMF
    end for
end function

4 PROPOSED MODIFIED PEER GROUP FUZZY METRIC FILTER (MPGFMF) FOR GPU ARCHITECTURES

Our peer group filter proposal was designed by executing only one kernel instead of two as the filter described in Section 3. The noise detection stage is used to know if the central pixel is a noisy one or not, if the central pixel is noisy, the filtering stage uses all the pixels within W to obtain a denoised one. The main idea behind this is to reduce the read/write count to the GDDR memory outperforming the computational cost from the work reported by Sanchez et al. [7, 8, 9], in addition, to improve the filtering stage, it is suggested to use the Median Filter (MF) with forgetful selection sort [17] applied for each color channel. The forgetful selection sort algorithm illustrated in Figure 4 consists of four iterations for a window processing W with $n = 3$ for each RGB channel; in the "Iteration 1", from the first six of the nine elements array the minimum and the maximum intensity values are obtained, then they are discarded of the array; in the "Iteration 2" the max and min values are computed again of the remaining four pixels adding the seventh element of the original array. These steps are repeated until the final "Iteration 4" where the mean pixel intensity is obtained value for every color channel. Along with these ideas, the proposed filter takes the advantage of the register memory that resides inside the GPU chip instead of the slowest GDDR memory [17, 8] used in the related work. In the kernel Algorithm 3, the proposed Modified Peer Group Fuzzy Metric Filter (MPGFMF) is shown. In Section 5, a memory cost analysis and a quality comparison between the proposal and reported filters in the literature are presented.

5 MEMORY COST ANALYSIS AND COMPARATIVE

A memory cost comparison between the state-of-the-art method PGMF and the proposed MPGFMF filtering algorithm considers the computational cost in the detection stage, so the number of accesses C_D is delivered in Equation (4), in the filtering stage, the computational cost is measured denoting C_{NF} (noise free) and C_N (noisy) described in Equation (5) and (6), respectively, considering that the PGMF filter computational analysis encompasses two cases where the pixel is noisy or free of noise.

$$C_D = \chi \cdot n^2, \tag{4}$$

$$C_{NF} = 1 + 2 \cdot \chi, \text{ noise-free pixel,} \tag{5}$$

$$C_N = 1 + (n^2 - 1) + \chi \cdot (n^2 - 1) + \chi, \text{ noisy pixel.} \tag{6}$$

For the PGMF method, $n = 3$ and $\chi = 3$ are taken, the noise-free case requires 35 reading and writing accesses for detection and filtering, and the noisy pixel requires 61 reading and writing accesses for the detection and filtering, both for the GDDRAM accesses. For our proposal, the same parameter values for n and χ are proposed resulting in 30 reading and writing accesses for the detection and filtering

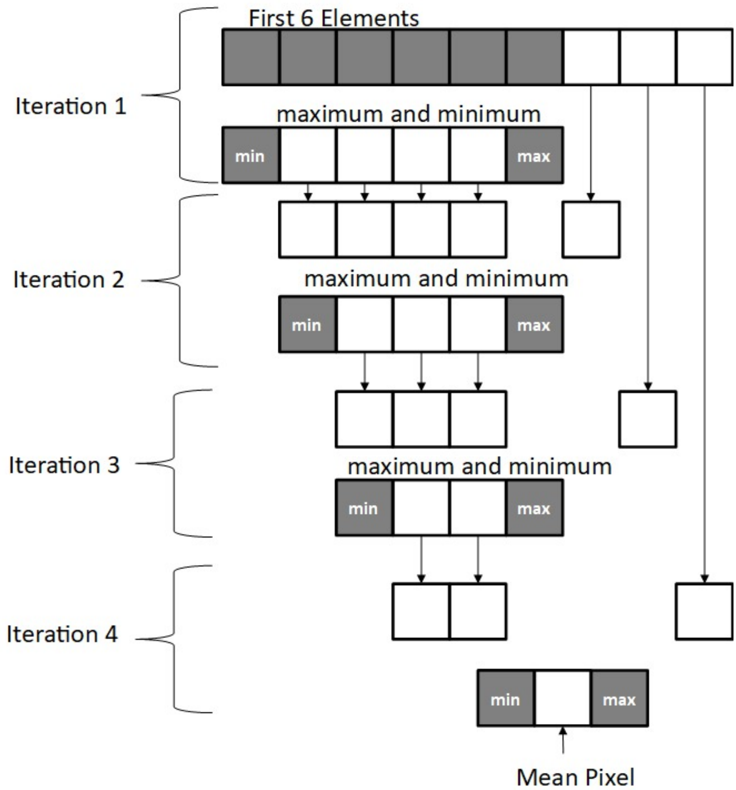


Figure 4. Forgetful selection algorithm

stage, also for the GDDRAM accesses. All the pixels within W for the proposed filtering algorithm are considered. Carrying out the count of readings and writings to the GDDR memory outperforms other parallel filters. The detection stage of the proposed filter needs:

$$C_D = \chi \cdot n^2. \tag{7}$$

For the filtering stage, it is proposed to reuse the memory in the detection stage, because one kernel is used and all the used variables in the present thread are still available for us. The difference from the PGMF filter which has two kernels and local variables and registers is that the kernels are scope-only (from Table 1). So, the proposed filter requires the same amount of readings and writings accesses no matter if the pixel within W is noisy or not, only needing χ writings for the pixel resulting from the filtering process, computed as in Equation (8):

$$C_N = C_{NF} = \chi, \tag{8}$$

Algorithm 3 Proposed MPGFMF

function MPGFMF ▷ Input: noisyImage ▷ Output: filteredImage
  for For each image pixel P_{th} its corresponding processing thread th **do**
    $a_c^\chi \leftarrow \{R,G,B\}$ values of P_{th} from global memory
    for all the P_j^χ pixels within W **do**
      $b_j^\chi \leftarrow \{R,G,B\}$ values of P_j^χ
      distance = Fuzzymetric(a_c^χ, b_j^χ)
      if distance $\geq \alpha$ **then**
        pixel $b_j^\chi \in \mathcal{P}(P_c^\chi, \kappa, \alpha)$
      end if
    end for
    if $\#\mathcal{P}(P_c^\chi, \kappa, \alpha) \geq \kappa + 1$ **then**
      P_{th} is labeled as noise free pixel
    else
      P_{th} is labeled as noisy
    end if
    if P_{th} is labeled as noisy pixel **then**
      Filter P_{th} all the pixels within W using the MMF with forgetful selection sort
    else
      P_{th} is the output pixel
    end if
  end for
end function

when $n = 3$, and $\chi = 3$, requires 30 readings and writings to the GDDRAM for "noise-free" and "noisy" cases. The following section shows the experimental results of the proposal and the method used as a comparison.

6 EXPERIMENTAL RESULTS

Here are compared the filters found in the scientific literature against our proposal using objective parameters. The objective parameters are defined as Peak Signal to Noise Ratio (PSNR) [18]:

$$PSNR = 10 \cdot \log \left[\frac{255^2}{MSE} \right] \tag{9}$$

where

$$MSE = \frac{1}{3 \times M \times N} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} \left[\left(R_{x,y}^O - R_{x,y}^F\right)^2 + \left(G_{x,y}^O - G_{x,y}^F\right)^2 + \left(B_{x,y}^O - B_{x,y}^F\right)^2 \right] \tag{10}$$

where $R^O_{x,y}$, $G^O_{x,y}$ and $B^O_{x,y}$ are the color components of the original image and $R^F_{x,y}$, $G^F_{x,y}$ and $B^F_{x,y}$ are the color components of the filtered one.

The Mean Chromaticity Error (MCRE) [19] is defined as follows:

$$MCRE = \frac{1}{M \times N} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} \mu\left[O_{x,y}, F_{x,y}\right] \tag{11}$$

where $\mu[O_{x,y}, F_{x,y}]$ is the PP' distance in the Maxwell triangle between the pixel with coordinates x, y within the original image $O_{x,y}$ and the filtered pixel with coordinates x, y within the filtered one $F_{x,y}$.

The Normalized Color Difference (NCD) described in [20] is defined in Equation (12), where $L^O_{x,y}$, $u^O_{x,y}$, $v^O_{x,y}$ and $L^F_{x,y}$, $u^F_{x,y}$, $v^F_{x,y}$ are the values of the perceived lightness and two chrominance values related to $O_{x,y}$ and $F_{x,y}$, respectively.

$$NCD = \frac{\sum_{x=0}^{M-1} \sum_{y=0}^{N-1} \sqrt{(L^O_{x,y} - L^F_{x,y})^2 + (u^O_{x,y} - u^F_{x,y})^2 + (v^O_{x,y} - v^F_{x,y})^2}}{\sum_{x=0}^{M-1} \sum_{y=0}^{N-1} \sqrt{(L^O_{x,y})^2 + (u^O_{x,y})^2 + (v^O_{x,y})^2}}. \tag{12}$$

Structural Similarity [21] (SSIM), which characterizes the preservation of the structures of the image is given by:

$$SSIM = \frac{(2\mu_O \mu_F + C_1)(2\sigma_{OF} + C_2)}{(\mu_O^2 + \mu_F^2 + C_1)(\sigma_O^2 + \sigma_F^2 + C_2)} \tag{13}$$

where μ_O and μ_F are the averages of $O_{x,y}$ and $F_{x,y}$, respectively. σ_O^2 and σ_F^2 are the variances of the two images. σ_{OF} is the covariance of $O_{x,y}$ and $F_{x,y}$. $C_1 = (0.01L)^2$ and $C_2 = (0.03L)^2$ are two constants and $L = 2^8$ is the dynamic range of the pixel values.

The visual signal-to-noise ratio (VSNR) [22, 23] characterizes the Visual Fidelity of images taking into consideration the human visual system. These criteria consist of 3 stages, namely Preprocessing, Assess the Detectability of the Distortions and Compute the VSNR, where:

$$VSNR = 20 \cdot \log_{10}\left(\frac{C(O)}{\alpha d_{pc} + (1-\alpha)(d_{gp}/\sqrt{2})}\right) \tag{14}$$

where $C(O)$ is the contrast distortion of the original image, d_{pc} is the contrast perceived of the distortions obtained from the filtered image respect to the original one, d_{gp} is the disruption calculation of global precedence and α is a suggested weight, in our case $\alpha = (0.04)^4$ [22].

Comparison of the filtering methods is achieved using the PC computer – its characteristics are listed in Table 2.

The test color images used are the well known "Lena" and "Mandrill", shown in Figure 5 a) and 5 b), respectively, the spatial resolution size is of 512×512. These

Characteristics	Value	Characteristics	Value
CPU	Intel Core i7-8700K @ 3.70 GHz	GPU	Nvidia GeForce RTX 2080
CPU architecture	Coffee Lake	GPU architecture	Turing
Cache	12 Mb	SMs	46
Core count	6 cores	SPs in each SM	64
System Memory	16 Gb DDR4	Memory	8 Gb GDDR6
OS	Microsoft Windows 10, 64-bit		

Table 2. Characteristics of the system used in the CPU and GPU

images were resized from 256×256 using Matlab, leaving grayscale images with intensity values from 0 to 255, and with 24-bit color images per pixel. The resized versions of the images of $1\,024 \times 1\,024$, $2\,048 \times 2\,048$, $4\,096 \times 4\,096$ and $8\,192 \times 8\,192$ pixels were used for the runtime tests. We also tested the performance of the filters using a 3D and a 2D image shown in Figures 5 c) and 5 d), both 8 bit grayscale images. The 3D grayscale image is a Magnetic Resonance Imaging (MRI) scan, with dimensions $256 \times 320 \times 192$ and was obtained from the dataset of I Do Imaging (file 1010_brain_mr_06.nii.gz) found in [24] in the Neuroimaging Informatics Technology Initiative NIfTI-1 Data Format. We used the Digital Imaging and Communications in Medicine (DICOM) Converter version: 1.10.5 from DICOM apps in order to obtain bitmap images for testing. For the PSNR, SSIM and VSNR experiments, we used the image number 96 of the 3D set. The 2D grayscale image is a computed tomography (CT) mammography scan of size $1\,024 \times 1\,024$ obtained from the mini-MIAS database of mammograms (file mdb001.pgm) [25]. Furthermore, the filter proposal was tested against a state-of-the-art Deep Learning Filter (DLF) for impulse noise found in [26]. The DLF methods are very promising techniques to be used in the digital image denoising, the efforts until now are focused in grayscale denoising because a lot of computational resources needs to be used. The quality results obtained from DLF are impressive, but they spend a lot of computational time, as shown in [26, 27]. The results for the PSNR and SSIM for the test grayscale images of 512×512 pixels are shown for the Lena, Barbara, Boat, and 256×256 Cameraman. Besides, the runtime is given as a criterion for the GPU.

7 DISCUSSION

It is demonstrated that our proposal has better numerical results compared with the other state-of-the-art GPU filters for the Mandrill image in all the noise density interval. In Tables 3 and 4 there are shown the denoising results of the Mandrill image corrupted with Random-value and Salt and Pepper impulsive noises from 0 through 40 %, with incremental steps of 5 %; evaluation criteria used are PSNR, MCRE, NCD, SSIM, and VSNR. It is possible to see that our filter outperforms in

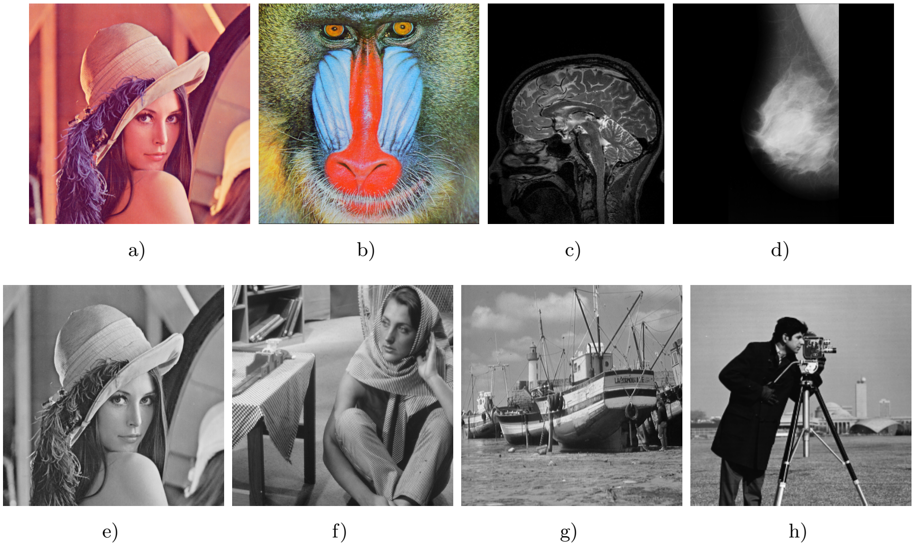


Figure 5. Digital Images used: a) Lena, b) Mandril c) Brain, d) Mammography, e) Lena Grayscale, f) Barbara, g) Boat, and h) Cameraman

the whole noise interval (0 to 40 %). As the noise density increases, it is possible to note that the difference in the numerical results increases, i.e. in Table 4, for 0 % $PGMF_{VMF}$ has a PSNR = 32.33 dB, $PGMF_{AMF}$ has a PSNR = 32.6 dB and our filter 32.71 dB, for 40 % we can see that $PGMF_{VMF}$ has a PSNR = 18.44 dB, $PGMF_{VMF}$ has a PSNR = 16.68 dB and our filter 19.51 dB.

In Table 5, the PSNR, MCRE, NCD, SSIM, and VSNR results concerning the Lena image corrupted with Random-value and Salt and Pepper impulsive noise from 0 through 40 % with steps of 5 % are presented. You can see in Table 5 that the proposed filter has similar good performance in the PSNR, MCRE, NCD, SSIM and VSNR criteria, outperforming the GPU filters ($PGMF_{\mathrm{VMF}}$ and $PGMF_{\mathrm{AMF}}$), that is, for 10 % we obtained by means of our filter proposal PSNR values of 26.11 dB, while $PGMF_{\mathrm{VMF}} = 25.99$ dB and $PGMF_{\mathrm{AMF}} = 25.98$ dB. In Table 6, there are shown similar results, that is, for example, by our proposal for 10 % of Salt and Pepper noise a PSNR result of 29.26 dB is obtained, while for comparative filters $PGMF_{\mathrm{VMF}} = 25.38$ dB and $PGMF_{\mathrm{AMF}} = 25.42$ dB were obtained in PSNR criterion, and so on for all the other criteria implemented.

For the Brain image, the PSNR, SSIM, and VSNR criteria results for the proposal and comparative filters using Salt and Pepper and Random-value impulsive noise are presented. The proposal presents better results for the criteria from 0 through 40 % with steps of 5 % of noise density, as seen in Tables 7 and 8. In Table 7 the best results for the SSIM criterion can be perceived, for example, for 20 % of Random-value impulsive noise for MPGFMF the $SSIM = 0.571$ was obtained,

	PGMF$_{VMF}$					PGMF$_{AMF}$					MPGFMF				
%	PSNR	MCRE	NCD	SSIM	VSNR	PSNR	MCRE	NCD	SSIM	VSNR	PSNR	MCRE	NCD	SSIM	VSNR
0	32.33	0.143	0.000	0.971	12.459	32.6	0.127	0.000	.972	12.399	**32.71**	**0.117**	0.000	**0.987**	**24.017**
5	26.82	4.415	0.006	0.909	12.887	27.12	4.241	0.006	0.912	12.839	**27.37**	**4.239**	**0.005**	**0.930**	**22.318**
10	24.42	8.261	0.011	0.849	12.656	24.75	**7.967**	0.011	0.854	12.622	**25.05**	7.977	**0.010**	**0.875**	**20.485**
15	22.78	11.860	0.017	0.790	12.351	23.12	11.489	0.016	0.796	12.355	**23.46**	**11.475**	**0.015**	**0.822**	**18.559**
20	21.59	15.254	0.021	.735	11.815	21.9	14.850	0.021	0.742	11.824	**22.29**	**14.766**	**0.019**	**0.773**	**17.106**
25	20.66	18.426	0.026	0.684	11.392	21.00	17.979	0.026	0.693	11.212	**21.43**	**17.800**	**0.023**	**0.729**	**15.997**
30	19.81	21.492	0.031	0.636	10.806	20.15	20.934	0.031	0.646	10.588	**20.71**	**20.569**	**0.027**	**0.689**	**14.872**
35	19.09	24.556	0.035	0.589	10.292	19.39	23.846	0.03	0.598	9.956	**20.05**	**23.310**	**0.031**	**0.649**	**14.067**
40	18.44	27.459	0.039	0545	9.795	18.68	26.479	0.042	0.551	9.128	**19.51**	**25.840**	**0.034**	**0.614**	**13.132**

Table 3. Criteria results for the Mandril image with random-value impulsive noise

	PGMF$_{VMF}$					PGMF$_{AMF}$					MPGFMF				
%	PSNR	MCRE	NCD	SSIM	VSNR	PSNR	MCRE	NCD	SSIM	VSNR	PSNR	MCRE	NCD	SSIM	VSNR
0	32.33	0.143	0.000	0.971	12.459	32.6	0.127	0.000	0.972	12.399	**32.71**	**0.117**	0.000	**0.987**	**24.017**
5	26.16	4.306	0.007	0.901	12.779	26.65	3.963	0.007	0.907	12.765	**27.15**	**4.000**	**0.006**	**0.929**	**21.650**
10	22.81	9.0551	0.015	0.805	12.161	23.3	**8.500**	0.015	0.816	12.116	**24**	8.505	**0.013**	**0.852**	**18.725**
15	20.56	14.1079	0.024	0.717	11.204	20.94	13.406	0.024	0.717	10.874	**22.02**	**13.051**	**0.019**	**0.775**	**16.588**
20	18.64	20.363	0.035	0.596	10.121	18.64	19.077	0.041	0.596	9.039	**20.53**	**17.863**	**0.026**	**0.698**	**14.905**
25	16.81	28.405	0.047	0.483	9.176	16.31	25.439	0.007	0.455	6.866	**19.22**	**22.968**	**0.032**	**0.625**	**13.622**
30	15.13	38.836	0.064	0.379	8.166	14.05	31.293	0.129	0.318	4.607	**18.18**	**28.054**	**0.038**	**0.559**	**12.458**
35	13.44	52.492	0.086	0.279	7.278	11.95	36.951	0.231	0.198	2.894	**17.01**	**34.165**	**0.046**	**0.484**	**11.096**
40	12.03	66.670	0.670	0.204	6.667	10.37	40.575	0.378	0.120	2.168	**15.91**	**40.850**	**0.054**	**0.417**	**9.932**

Table 4. Criteria results for the Mandril image with Salt and Pepper impulsive noise

	$PGMF_{VMF}$					$PGMF_{AMF}$					MPGFMF				
%	PSNR	MCRE	NCD	SSIM	VSNR	PSNR	MCRE	NCD	SSIM	VSNR	PSNR	MCRE	NCD	SSIM	VSNR
0	29.25	0.006	0.000	0.963	13.251	29.26	0.006	0.000	.963	13.255	**29.26**	**0.004**	0.000	**0.983**	**21.11**
5	27.48	2.638	0.003	0.902	13.518	27.46	2.647	0.003	0.882	13.395	**27.54**	**2.570**	**0.003**	**0.901**	**21.49**
10	25.99	5.184	0.007	0.822	13.541	25.98	5.318	0.007	0.807	13.469	**26.11**	7.977	**.006**	**0.826**	**21.11**
15	24.63	7.665	0.107	0.724	13.455	24.63	8.043	0.011	0.733	13.409	**24.79**	**7.553**	**0.010**	**0.752**	**20.33**
20	23.55	9.969	0.014	0.611	13.017	23.57	10.652	0.014	0.673	13.249	**23.80**	**9.837**	**0.0133**	**0.693**	**19.61**
25	22.58	12.352	0.017	0.494	12.043	22.57	13.440	0.019	0.616	12.867	**21.43**	**12.238**	**0.167**	**0.637**	**18.68**
30	21.72	14.790	0.021	0.379	11.041	21.71	16.242	0.023	0.568	12.487	**22.05**	**14.668**	**0.0201**	**0.588**	**17.81**
35	20.96	17.139	0.024	0.266	9.624	20.92	18.859	0.027	0.521	12.000	**21.42**	**16.842**	**0.234**	**0.547**	**17.03**
40	20.19	19.770	0.028	0.180	8.363	20.10	21.577	0.032	0.479	11.310	**20.76**	**19.159**	**0.026**	**0.509**	**16.10**

Table 5. Criteria results for the Lena image with Random-value impulsive noise

	$PGMF_{VMF}$					$PGMF_{AMF}$					MPGFMF				
%	PSNR	MCRE	NCD	SSIM	VSNR	PSNR	MCRE	NCD	SSIM	VSNR	PSNR	MCRE	NCD	SSIM	VSNR
0	29.25	0.006	0.000	0.963	13.251	29.26	0.006	0.000	0.963	13.255	**29.26**	**0.004**	0.000	**0.983**	**21.11**
5	27.6	1.950	0.003	0.902	13.518	27.6	1.943	0.003	0.903	13.395	**27.73**	**1.829**	**0.003**	**0.901**	**21.49**
10	25.38	4.358	0.007	0.822	13.541	25.42	4.565	0.008	0.823	13.469	**25.68**	4.114	**0.006**	**0.826**	**21.11**
15	23.07	7.560	0.013	0.724	13.455	23.12	8.260	0.013	0.724	13.409	**23.58**	**7.002**	**0.010**	**0.752**	**20.33**
20	20.87	11.994	0.020	0.611	13.017	20.87	13.197	0.022	0.606	13.249	**21.83**	**10.354**	**0.015**	**0.693**	**19.61**
25	18.89	17.956	0.029	0.429	12.043	18.58	18.814	0.039	0.473	12.867	**20.53**	**13.764**	**0.020**	**0.637**	**18.68**
30	16.96	25.877	0.043	0.379	11.041	16.16	24.897	0.071	0.339	12.487	**19.34**	**17.498**	**0.024**	**0.588**	**17.81**
35	15.00	37.424	0.063	0.266	9.624	13.69	31.921	0.135	0.217	12.001	**18.07**	**22.368**	**0.030**	**0.547**	**17.03**
40	13.33	50.593	0.090	0.179	8.363	11.60	37.928	0.243	0.130	11.310	**16.85**	**28.273**	**0.038**	**0.509**	**16.10**

Table 6. Criteria results for the Lena image with Salt and Pepper impulsive noise

and for comparative ones, for $\mathrm{PGMF_{VMF}}$ the $SSIM = 0.538$ and for $\mathrm{PGMF_{AMF}}$ the $SSIM = 0.563$. And also, for the results in Table 8, for 20 % of Salt and Pepper impulsive noise for MPGFMF the $SSIM = 0.571$ is obtained, and for comparative ones, for $\mathrm{PGMF_{VMF}}$ the $SSIM = 0.539$ and for $\mathrm{PGMF_{AMF}}$ the $SSIM = 0.567$.

	$\mathrm{PGMF_{VMF}}$			$\mathrm{PGMF_{AMF}}$			MPGFMF		
%	PSNR	SSIM	VSNR	PSNR	SSIM	VSNR	PSNR	SSIM	VSNR
0	41.87	0.994	36.16	**42.76**	0.994	36.67	42.50	**0.994**	**37.40**
5	31.05	0.901	27.93	**31.55**	0.907	**28.53**	31.52	**0.907**	28.51
10	26.48	0.792	25.17	27.04	0.806	26.08	**27.08**	**0.808**	**26.34**
15	23.61	0.671	24.93	24.30	0.692	25.22	**24.35**	**0.694**	**25.27**
20	20.92	0.538	25.59	21.61	0.563	25.77	**21.69**	**0.566**	**26.04**
25	19.23	0.446	24.99	19.81	0.471	25.06	**19.89**	**0.474**	**25.34**
30	17.65	0.358	24.54	18.19	0.381	24.57	**18.27**	**0.384**	**24.88**
35	16.3	0.290	23.38	16.88	0.312	23.18	**16.95**	**0.314**	**23.46**
40	15.24	0.258	**24.84**	15.77	0.259	24.45	**15.81**	**0.261**	24.63

Table 7. Criteria results for the Brain image with Random-value impulsive noise

	$\mathrm{PGMF_{VMF}}$			$\mathrm{PGMF_{AMF}}$			MPGFMF		
%	PSNR	SSIM	VSNR	PSNR	SSIM	VSNR	PSNR	SSIM	VSNR
0	41.87	0.994	36.16	**42.76**	0.994	36.67	42.50	**0.994**	**37.40**
5	30.28	0.942	24.45	31.18	0.950	25.34	**31.19**	**0.950**	**28.55**
10	24.15	0.846	22.37	24.84	0.862	23.27	**24.88**	**0.863**	**23.16**
15	20.50	0.706	21.49	21.03	0.727	22.00	**21.10**	**0.729**	**21.87**
20	17.22	0.539	22.01	17.76	0.567	22.17	**17.82**	**0.571**	**22.21**
25	15.03	0.383	21.17	15.47	0.408	21.62	**15.53**	**0.412**	**21.58**
30	13.16	0.264	20.99	13.54	0.285	21.26	**13.61**	**0.290**	**21.41**
35	11.74	0.176	20.95	12.09	0.192	21.21	**12.16**	**0.196**	**21.27**
40	10.40	0.115	20.71	10.71	0.127	20.93	**10.77**	**0.130**	**21.19**

Table 8. Criteria results for the Brain image with Salt and Pepper impulsive noise

Tables 9 and 10 show the results for the Mammography image. Our filter presents the best results for the criteria implemented from 0 through 40 % with steps of 5 % of noise density. In Table 9, results for the Brain image corrupted with Random-value impulse noise are presented, the proposal MPGFMF for the VSNR results for 30 % of noise density is of $VSNR = 25.12$, while for the comparative ones, $\mathrm{PGMF_{VMF}}$ presents $VSNR = 24.75$, and for $\mathrm{PGMF_{AMF}}$ has $VSNR = 23.05$. The same tendency is perceived in Table 10 for all the criteria implemented and for all the noise percentage levels, showing that our proposal outperforms the comparative ones.

In Figure 6 there is presented a visual comparison between the three filters using a zoomed region of Lena image corrupted with 5 %, 15 % and 25 % of Salt and Pepper impulsive noise. It is denoted that for 5 % and 15 % the differences between the three

	$PGMF_{VMF}$			$PGMF_{AMF}$			MPGFMF		
%	PSNR	SSIM	VSNR	PSNR	SSIM	VSNR	PSNR	SSIM	VSNR
0	26.01	0.978	48.41	26.01	0.978	48.41	**26.01**	**0.978**	**48.41**
5	25.35	0.841	41.85	25.39	0.843	41.85	**25.40**	**0.844**	42.31
10	23.908	0.689	35.57	24.09	0.697	35.57	**24.12**	**0.700**	**36.16**
15	22.00	0.523	31.69	22.35	0.535	31.69	**22.39**	**0.541**	**32.20**
20	20.05	0.370	28.27	20.48	0.384	28.27	**20.52**	**0.391**	**28.99**
25	18.39	0.254	26.16	18.86	0.268	24.75	**18.91**	**0.273**	**26.51**
30	16.95	0.167	24.75	17.44	0.178	23.05	**17.48**	**0.183**	**25.12**
35	15.71	0.112	23.05	16.21	0.1212	22.14	**16.24**	**0.124**	**23.53**
40	14.72	0.077	22.14	15.20	0.084	22.13	**15.22**	**0.086**	**22.44**

Table 9. Criteria results for the Mammography image with Random-value impulsive noise

	$PGMF_{VMF}$			$PGMF_{AMF}$			MPGFMF		
%	PSNR	SSIM	VSNR	PSNR	SSIM	VSNR	PSNR	SSIM	VSNR
0	26.01	0.978	48.41	26.01	0.978	48.41	**26.01**	**0.978**	**48.41**
5	25.21	0.931	33.63	25.30	0.931	34.03	25.30	**0.933**	**33.98**
10	22.585	0.818	26.36	22.86	0.824	27.05	**22.89**	**0.826**	**27.21**
15	19.45	0.635	21.76	19.82	0.635	22.36	**19.86**	**0.650**	**22.26**
20	16.79	0.438	19.42	17.16	0.438	19.81	**17.20**	**0.456**	**19.76**
25	14.56	0.263	17.64	14.90	0.263	18.09	**14.94**	**0.280**	**18.21**
30	12.85	0.144	16.78	13.17	0.144	17.04	**13.21**	**0.151**	**17.08**
35	11.37	0.074	16.02	11.67	0.074	16.24	**11.71**	**0.083**	**16.31**
40	10.17	0.038	15.41	10.43	0.038	15.71	**10.46**	**0.046**	**15.81**

Table 10. Criteria results for the Mammography image with Salt and Pepper impulsive noise

filters are imperceptible only with numerical results. However, in the case of 25 % of impulsive noise, the proposal filter presents better noise suppression quality in visual representation denoted by the noisy clusters zones. The $PGMF_{AMF}$ filter propagates the noise between the pixels, this way the $PGMF_{VMF}$ filter does not suppress the noisy pixels formed in clusters well enough compared to our filter proposal. Our filter proposal (MPGFMF) does not propagate the noise and the noisy pixels in clusters are processed in a better way improving performance. Table 13 shows the computational cost performance of the three filters using the Lena image with resolutions of 512×512, $1\,024 \times 1\,024$, $2\,048 \times 2\,048$, $4\,096 \times 4\,096$ and $8\,192 \times 8\,192$. Superior filtering quality and fast execution in the proposal filter is due to changes proposed in the detection and filtering stages, the algorithm decreases the reading and writing counts, dealing better with noisy pixels in consequence of pixel clusters. The forgetful selection algorithm requires low register accesses inside the SMs and does not depend on the magnitude calculations between color pixels, performing better than the Vector Median Filter.

Next, in Table 14 a comparison with the state-of-the-art Deep Learning Filter (DLF) [26] for Random-value impulse noise for PSNR and SSIM for noise densities

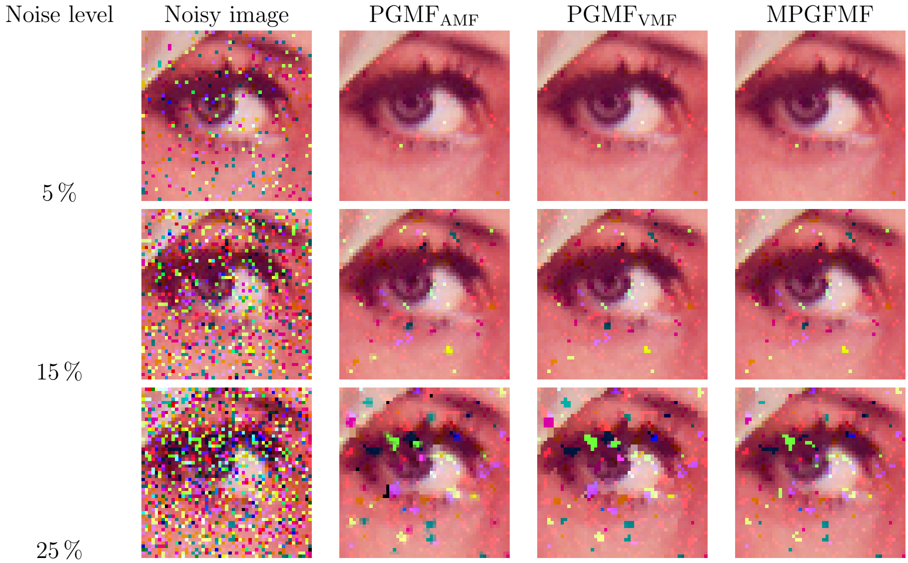


Figure 6. Visual comparative results between algorithms studied

Image Size	Filter	Time (ms)	Image Size	Filter	Time (ms)
512×512	$\mathrm{PGMF_{AMF}}$	0.739	$4\,096 \times 4\,096$	$\mathrm{PGMF_{AMF}}$	38.408
	$\mathrm{PGMF_{VMF}}$	1.730		$\mathrm{PGMF_{VMF}}$	86.419
	MPGFMF	0.841		MPGFMF	41.568
$1\,024 \times 1\,024$	$\mathrm{PGMF_{AMF}}$	2.807	$8\,192 \times 8\,192$	$\mathrm{PGMF_{AMF}}$	153.944
	$\mathrm{PGMF_{VMF}}$	6.594		$\mathrm{PGMF_{VMF}}$	348.540
	MPGFMF	3.172		MPGFMF	160.628
$2\,048 \times 2\,048$	$\mathrm{PGMF_{AMF}}$	9.422			
	$\mathrm{PGMF_{VMF}}$	21.731			
	MPGFMF	10.213			

Table 11. Runtime of the three filters using the Lena image corrupted with 10 % Salt and Pepper impulsive noise

10, 20, 30, 40, 50 and 60 % is made. From Table 14 we can perceive the poor results compared to a DLF filter, in PSNR and SSIM terms for 10, 20, 30, 40, 50 and 60 %.

For the runtime tests, as in [26], we used Cameraman (256×256), Sailing Boats (512×512) and Lena (512×512) corrupted with 10 %, 30 % and 50 % of Random-value impulse noise. In [26], the authors used an Intel i7 CPU based system and an Nvidia GeForce 960 GPU. Their obtained results and our results for the same images are shown in Table 15. Table 15 shows that our filter outperforms the Deep Learning filtering algorithm, i.e., for the Lena image with 10 % the runtime of filter found in [26] is 0.97 seconds. Our filter proposal has a running time of 0.0003 seconds. These results lead to a speed-up of 3 233.33 times.

Image Size	Filter	Time (ms)	Image Size	Filter	Time (ms)
$256 \times 320 \times 192$	PGMF_{AMF}	20.544	$1\,024 \times 1\,024$	PGMF_{AMF}	0.769
	PGMF_{VMF}	55.296		PGMF_{VMF}	0.763
	MPGFMF	19.968		MPGFMF	0.713

Table 12. Runtime of the three filters using the MRI Brain 3D image (image number 96) and the CT Mammography with 10 % of Salt and Pepper impulse noise

Image Size	Filter	Time (ms)	Image Size	Filter	Time (ms)
512×512	PGMF_{AMF}	0.739	$4\,096 \times 4\,096$	PGMF_{AMF}	38.408
	PGMF_{VMF}	1.730		PGMF_{VMF}	86.419
	MPGFMF	0.841		MPGFMF	41.568
$1\,024 \times 1\,024$	PGMF_{AMF}	2.807	$8\,192 \times 8\,192$	PGMF_{AMF}	153.944
	PGMF_{VMF}	6.594		PGMF_{VMF}	348.540
	MPGFMF	3.172		MPGFMF	160.628
$2\,048 \times 2\,048$	PGMF_{AMF}	9.422			
	PGMF_{VMF}	21.731			
	MPGFMF	10.213			

Table 13. Runtime of the three filters using the Lena image corrupted with 10 % Salt and Pepper impulsive noise

In our literature review, we found that little work has been done regarding image color filtering of impulse noise in parallel systems. We refocus previous ideas found in the literature, i.e. Peer Group Filtering and Median filtering with forgetful selection sort for grey-scale images, in order to improve performance of impulse noise filtering of color images further. In all experimental intervals we obtained better numerical results and qualitative results. We found that it is possible to run Parallel Peer Group Filters in real-time (< 33 ms per frame) up to $1\,024 \times 1\,024$ pixels in an Nvidia 2080 RTX card.

Criterion	PSNR (dB)						SSIM (r.u.)					
Noise (%)	10	20	30	40	50	60	10	20	30	40	50	60
Lena												
DLF	43.47	40.73	38.57	36.5	33.98	30.46	0.985	0.975	0.963	0.948	0.927	0.886
MPGFMF	27.67	26.87	25.97	24.7	23.36	22.12	0.823	0.765	0.718	0.658	0.588	0.517
Cameraman												
DLF	38.88	35.42	32.9	30.59	28.52	26.06	0.977	0.956	0.934	0.908	0.878	0.826
MPGFMF	24.34	23.5	22.6	21.38	20.22	19.01	0.773	0.702	0.642	0.555	0.483	0.418
Barbara												
DLF	43.44	40.14	37.61	35.51	33.2	30.82	0.990	0.978	0.959	0.932	0.878	0.797
MPGFMF	23.53	22.92	22.27	21.51	20.22	19.01	0.766	0.704	0.650	0.583	0.512	0.443
Boat												
DLF	42.51	39.58	37.56	35.86	34.22	32.48	0.973	0.953	0.931	0.904	0.866	0.800
MPGFMF	27.67	26.87	25.97	24.7	23.36	22.12	0.824	0.753	0.698	0.635	0.570	0.506

Table 14. PSNR and SSIM results applied by Lena, Cameraman, Barbara and Boat images corrupted by 10, 20, 30, 40, 50 and 60 % Random-value impulsive noise

Images	Cameraman			Sailing Boats			Lena		
Noise Density (%)	10	30	50	10	30	50	10	30	50
DLF	0.24	0.23	0.23	1.05	0.97	0.97	0.97	0.97	0.97
MPGFMF	0.000103	0.000103	0.000103	0.0003	0.0003	0.0003	0.0003	0.0003	0.0003

Table 15. Runtime (secs.) comparison of a Deep Learning algorithm [26] against our proposal by 10, 30, and 50 % of Random-value impulsive noise

8 CONCLUSIONS

A novel GPU filtering algorithm for impulsive denoising applied to color images was proposed presenting better performance in preserving edges, details, structures, and chromaticity properties than those used as comparative. The criteria used to provide these validation quantitative results show that the *MPGFMF* preserves better the properties inherent to the images in suppressing the Random-value and Salt and Pepper impulsive noises. The execution runtime shows that the proposal is the second-fastest impulsive noise filter, comparing its rendering respect to the fastest PGMF_{AMF}, this filter is the worst in impulsive denoising, denoting that our proposal is the best option to denoise suppression and computational cost. For the MRI and CT images we found that our filter performs slightly better than the other GPU filters and for these types of images, our filter is the fastest. Compared with a state-of-the-art Deep Learning impulse noise filter our filter is not good in PSNR and SSIM terms for grayscale images, but in execution time, our filter is much faster ($> 3\,000$ times). To further improve the proposed filter, it is necessary to study and implement existing ordering algorithms or design a new one. We found that little work has been done regarding parallel color filtering and some research is needed, i.e. 3D color filtering of video sequences.

Acknowledgements

The authors are grateful to the Instituto Politécnico Nacional de México and the CONACyT de México for their support to this research work.

REFERENCES

[1] Salomon, C.—Breckon, T.: Fundamentals of Digital Image Processing. Wiley-Blackwell, 2011.

[2] Ponomaryov, V.—Montenegro, H.—Rosales, A.—Duchen, G.: Fuzzy 3D Filter for Color Video Sequences Contaminated by Impulsive Noise. Journal of Real-Time Image Processing, Vol. 10, 2015, pp. 313–328, doi: 10.1007/s11554-012-0262-9.

[3] Lukac, R.—Smolka, B.—Martin, K.—Plataniotis, K. N.—Venetsanopoulos, A. N.: Vector Filtering for Color Imaging. IEEE Signal Processing Magazine, Vol. 22, 2005, No. 1, pp. 74–86, doi: 10.1109/MSP.2005.1407717.

[4] Schulte, S.—De Witte, V.—Nachtegael, M.—Van Der Weken, D.—Kerre, E. E.: Fuzzy Two-Step Filter for Impulse Noise Reduction From Color Images. IEEE Transactions on Image Processing, Vol. 15, 2006, No. 11, pp. 3567–3578, doi: 10.1109/TIP.2006.877494.

[5] Owens, J. D.—Houston, M.—Luebke, D.—Green, S.—Stone, J. E.—Phillips, J. C.: GPU Computing. Proceedings of the IEEE, Vol. 96, 2008, No. 5, pp. 879–899, doi: 10.1109/JPROC.2008.917757.

[6] Smolka, B.: Fast Impulsive Noise Removal in Color Images. IEEE International Conference on Image Processing, 2013, pp. 1212–1216, doi: 10.1109/ICIP.2013.6738250.

[7] Sánchez, M. G.—Vidal, V.—Arnal, J.—Vidal, A.: Image Noise Removal on Heterogeneous CPU-GPU Configurations. Procedia Computer Science, Vol. 29, 2014, pp. 2219–2229, doi: 10.1016/j.procs.2014.05.207.

[8] Sánchez, M. G.—Vidal, V.—Bataller, J.—Arnal, J.: A Parallel Method for Impulsive Image Noise Removal on Hybrid CPU/GPU Systems. Procedia Computer Science, Vol. 18, 2013, pp. 2504–2507, doi: 10.1016/j.procs.2013.05.429.

[9] Sanchez, M. G.—Vidal, V.—Bataller, J.: Peer Group and Fuzzy Metric to Remove Noise in Images Using Heterogeneous Computing. In: Alexander, M. et al. (Eds.): Euro-Par 2011: Parallel Processing Workshops. Springer, Berlin, Heidelberg, Lecture Notes in Computer Science, Vol. 7155, 2011, pp. 502–510, doi: 10.1007/978-3-642-29737-3_55.

[10] Cook, S.: CUDA Programming: A Developer's Guide to Parallel Computing with GPUs. Morgan Kaufmann, Amsterdam, Boston, 2013, doi: 10.1016/C2011-0-00029-7.

[11] Tuning CUDA Applications for Maxwell. `http://docs.nvidia.com/cuda/maxwell-tuning-guide/index.html`.

[12] Kirk, D. B.—Hwu, W. W.: Programming Massively Parallel Processors: A Hands-on Approach. 1st Edition. Morgan Kaufmann, Burlington, MA, 2010.

[13] Couturier, R.: Designing Scientific Applications on GPUs. Chapman and Hall, Boca Raton, Florida, 2013.

[14] CUDA C Programming Guide. Available at: `http://docs.nvidia.com/cuda/cuda-c-programming-guide/index.html`.

[15] Smolka, B.: Adaptive Truncated Vector Median Filter. IEEE International Conference on Computer Science and Automation Engineering, Vol. 4, 2011, pp. 261–266, doi: 10.1109/CSAE.2011.5952847.

[16] Morillas, S.—Gregori, V.—Peris-Fajarnés, G.—Latorre, P.: A Fast Impulsive Noise Color Image Filter Using Fuzzy Metrics. Real-Time Imaging, Vol. 11, 2005, No. 5-6, pp. 417–428, doi: 10.1016/j.rti.2005.06.007.

[17] Perrot, G.—Domas, S.—Couturier, R.: Fine-Tuned High-Speed Implementation of a GPU-Based Median Filter. Journal of Signal Processing Systems, Vol. 75, 2014, No. 3, pp. 185–190, doi: 10.1007/s11265-013-0799-2.

[18] Bovik, A.: Handbook of Image and Video Processing. Academic Press, 2005, doi: 10.1016/B978-0-12-119792-6.X5062-1.

[19] Trahanias, P. E.—Karakos, D.—Venetsanopoulos, A. N.: Directional Processing of Color Images: Theory and Experimental Results. IEEE Transactions on Image Processing, Vol. 5, 1996, No. 6, pp. 868–880, doi: 10.1109/83.503905.

[20] Lukac, R.: Adaptive Vector Median Filtering. Pattern Recognition Letters, Vol. 24, 2003, pp. 1889–1899, No. 12, doi: 10.1016/S0167-8655(03)00016-3.

[21] Wang, Z.—Bovik, A. C.—Sheikh, H. R.—Simoncelli, E. P.: Image Quality Assessment: From Error Visibility to Structural Similarity. IEEE Transactions on Image Processing, Vol. 13, 2004, No. 4, pp. 600–612, doi: 10.1109/TIP.2003.819861.

[22] Chandler, D. M.—Hemami, S. S.: VSNR: A Wavelet-Based Visual Signal-to-Noise Ratio for Natural Images. IEEE Transactions on Image Processing, Vol. 16, 2007, No. 9, pp. 2284–2298, doi: 10.1109/TIP.2007.901820.

[23] Poynton, C.: The Rehabilitation of Gamma. In: Rogowitz, B. E., Pappas, T. N. (Eds.): Human Vision and Electronic Imaging III. Proceedings of the SPIE, Vol. 3299, 1998, pp. 232–249.

[24] I Do Imaging. `https://wiki.idoimaging.com/index.php?title=Sample_Data`, accessed on 11/1/2019.

[25] The Mini-MIAS Database of Mammograms. `http://peipa.essex.ac.uk/info/mias.html`, accessed on 11/1/2019.

[26] Jin, L.—Zhang, W.—Ma, G.—Song, E.: Learning Deep CNNs for Impulse Noise Removal in Images. Journal of Visual Communication and Image Representation, Vol. 62, 2019, pp. 193–205, doi: 10.1016/j.jvcir.2019.05.005.

[27] Turkmen, I.: The ANN Based Detector to Remove Random-Valued Impulse Noise in Images. Journal of Visual Communication and Image Representation, Vol. 34, 2016, pp. 28–36, doi: 10.1016/j.jvcir.2015.10.011.

José Agustín Tortolero Osuna studies in the Instituto Politécnico Nacional de México. He is currently pursuing his Ph.D. degree focusing on image processing. His current interests are high-performance computing and image enhancement.

Alberto Jorge Rosales Silva works in the Instituto Politécnico Nacional de México. He obtained his Ph.D. in communications and electronics (2008), Master of Science in telecommunications engineering (2004) and Bachelors degree in engineer communications and electronics (1999) from ESIME-Culhuacan in Instituto Politécnico Nacional de México. Topics currently being developed in image processing, multispectral and multichannel video in real-time, visual computing, as well as medical image processing.

Computing and Informatics, Vol. 38, 2019, 1341–1374, doi: 10.31577/cai_2019_5_1341

AUTOMATIC QUERY REFINING BASED ON EYE-TRACKING FEEDBACK

Alena Martonova, Jozef Marcin, Pavol Navrat
Jozef Tvarozek, Gabriela Grmanova

Faculty of Informatics and Information Technologies
Slovak University of Technology in Bratislava
Ilkovičova 2, 842 16 Bratislava, Slovakia
e-mail: alena.martonova@stuba.sk

Abstract. This paper presents a new method named AQueReBET, which automatically refines a query set by an information seeker searching on the web. A revelation of the intention of an information seeker who is running a search can bring a significant improvement to the search process and to browsing as well. It is practically impossible to acquire such intention by the explicit indication (feedback) due to the fact that web browsing takes place in real time. Therefore the intention must be determined in some other way. We hypothesize that it can be approximated by means of the implicit feedback preferably in the form of data from an eye tracker and mouse. We propose a method which automatically refines a seeker's search query, and thus we can offer documents with higher relevance, decrease the number of query reformulations and increase the seeker's satisfaction. The query refinement is based on an analysis of gaze data from an eye tracker and on groupization. In the proposed method, we calculate word-level importance based on term frequency, term uniqueness (tf-idf) and total fixation duration within the subdocument (word's snippet in search results).

Keywords: Web search, query refinement, eye-tracking, groupization, implicit feedback

Mathematics Subject Classification 2010: 68-U35, 68-M11

1 INTRODUCTION

Nowadays, most information seekers rely on the web when searching for documents to find the desired information. However, typical web search engines expect text queries, which are rather imperfect ways of expressing the intention of an information seeker. Prompting the information seeker for additional information beyond the query itself could be seen as asking too much. Therefore, one of the current research challenges is to derive as much useful information as possible from so called implicit feedback, which can be collected unobtrusively [17] – based on an information seeker's interaction with the search engine. There are various kinds of implicitly provided data that information seekers generate while interacting, such as their choice of a particular snippet to read in more detail or even a track of their gaze. In this paper, we investigate how an information seeker's eye gaze data acquired from an eye-tracking device can be used for refining the seeker's query during his/her searching session. An eye tracker provides potentially a very comprehensive immediate feedback unobtrusively, without any explicit questions that often tend to be perceived as annoying and also without demanding the seekers to perform searching in some specific (artificial) way. Experimental results [15] show that eye tracking is a valuable real-time implicit source of information about what the user is searching for and that it can be used for real-time user interface adaptation.

The initial query is often reformulated during a typical web search. Approximately one third of all formulated queries are composed by gradually reformulating an initial query [31, 4]. The relevance of the initially received documents could be quite low, implying the necessity of explicit query reformulation by the information seekers themselves.

The aim of our research is to improve search outcomes by reducing the need for explicit query reformulation and increasing relevancy of the offered documents. We attempt to achieve this objective by acquiring and utilizing data obtained from an eye-tracker and using it to implicitly reformulate the query. Furthermore, we propose to complement this by so-called groupization (i.e. the data are used also from other previous users with the same or very similar search intention) that can provide useful additional information to interpret an information seeker's intention.

Our approach is based on the assumption that if we recognize the web seeker's intention, we should be able to offer more relevant documents in response to the initial query. Since we are not able to detect exactly the line the seeker is reading by eye tracking, we focused on larger areas such as search result snippets (snippet is a web-page element, which contains a short text representing one of the results in search engine result page). We detect the snippet on which the seeker fixates his/her gaze within a search engine's results page (hereinafter SERP). We extracted specific words out of these snippets, which can possibly give us a strong clue on how to refine the initial search query to get more relevant documents to his/her search intention.

The rest of the paper is structured as follows. In the next section, we present important related results as published in the current literature. Section 3 presents our proposed method for refining an information seeker's query based on the analysis of his/her gaze movements and enhanced also with the groupization method. Experiments are presented and discussed in Section 4. The paper concludes in Section 5, where suggestions of future work are briefly discussed.

2 RELATED WORK

Query refinement [7, 5] and groupization methods play a key role in our work. Let us take a look at some current and related approaches in this area. Many existing works are focused on how the information seekers work with the SERP and how they select the relevant documents. It was already established that seekers are not very keen to provide explicit feedback [27], e.g. in the form of some kind of a relevance feedback mechanism or explicit query reformulation. Instead, various forms of implicit feedback are to be preferred. One direction of research is to use mouse-clicking data as implicit feedback to reformulate a query [13]. The gaze of seekers who reformulate a query was studied by Eickhoff et al. [8] and Umemoto et al. [32]. Li et al. [19] tracked the gaze to acquire relevance of retrieved images to be used for query expansion in an image retrieval system. However, we focus on text documents in the present research, but note that studies have emerged recently that broaden the applicability of eye gaze analysis and enhance the analysis itself by incorporating also e.g. pupil dilatation [21]. A seeker's gaze as a source of implicit feedback for information retrieval is a topic of research [6, 29]. Granka et al. [10] explored the behaviour of users during web searches and established the minimum fixation duration for web searches, which represents the minimum time the seeker is looking at relevant information. It was experimentally defined as 200 to 300 milliseconds. Various methods how to estimate an information seeker's search intention based on eye-tracking data were explored [31]. They represent search intention in a table created from a set of pairs – term and its weight. Weights of terms (TermScore) were computed using various functions combining quantities such as the number of times that a seeker looked at a term and the term frequency. Others investigated whether word relevance to a seeker's current intent could be inferred from the text and his/her eye movements [20]. It has already been shown that gaze-based feedback can be used to expand a query [5]. A particular simple data acquired from eye tracking, i.e. attention time, was used to recommend new online items [34].

Research led by Buscher [5, 6] is the most similar to ours, both considering the method and the way it is evaluated. Our method differs mainly by the choice of tf-idf (see below) as the base formula that is gradually expanded. On the other hand, our approach was to make additional emphasis on term frequency combined with total fixation duration on the snippet (Equation (3)).

White et al. [30] addressed methods of groupization. They were looking for seekers with similar features to obtain more relevant links to their queries. Current trends in devising groups are as follows:

- similarity based on link clicks: determined by three ways:
  1. match the URL clicked on,
  2. match the domain of the URL clicked on, and
  3. consistency between the categories of topics of the URL clicked on;
- syntactic similarity – similarity of the querying (keywords);
- semantic similarity – even if the queries are not similar based on syntactic similarity, they can be similar in their meanings.

Another area of active research in which researchers are engaged, is obtaining text via eye tracking. Methods for extracting text have been explored by Biedert et al. [3]. They are working on the interesting specific problem of adjusting eye tracking errors when reading a structured text to automatically position the cursor at a proper place. We decided not to follow this line of research, since our aim is different.

Eye fixation coordinates are correlated with mouse cursor positions thus facilitating considerations of various behavioural patterns – reading, hesitation, scrolling, clicking [11, 26]. Therefore, it may very well be possible that some of the results obtained by a method employing eye gaze tracking, and particularly by the proposed method, could be achieved or at least approximated by using mouse cursor data, which is more accessible for commercial search engines. On the other hand, one of their conclusions claiming that the cursor approximates the gaze is misguided. Other than that of the mouse cursor, eye tracker provides data on the actual viewed location. In our research, we attempt to make use of this additional data to achieve new insights in the automatic query refinement.

3 PROPOSED METHOD FOR REFINING

Our method – automatic query refining based on eye-tracking feedback (AQueReBET) – primarily deals with word table creation (a similar data structure is sometimes referred to as an "intention vector" [25], which however could be misleading, since a vector is conventionally considered to be a one-dimensional structure). The table is created by selecting appropriate words from useful elements of pages (mostly snippet areas, see Figure 2) scanned by the gaze during a seeker's web search. Such a table can facilitate refining the initial query by better (i.e., more relevant) words and thus offer the seeker results, which better fit the seeker's intention. Our hypothesis is: Some of the information obtained by tracking a seeker's gaze, can, after suitable processing, be helpful in providing search results that better reflect their intention. Unlike the methods based on mouse tracking, our method does not need to wait until the seeker moves the mouse cursor within the open page, the seeker clicks within the open page, the seeker opens any other page, or the seeker chooses some additional words to pose a new query.

While devising such a method, three assumptions emerged:

1. The longer the user's fixation within the snippet area, the higher the probability that it is closer to his/her intention than other snippets (the strongest criterion). Here, we have been inspired by Maglio et al. [22] who use eye-gaze information to help disambiguate user interests.
2. The higher the term frequency in watched snippets, the higher the probability that the term is closer to the user's intention. Here we have been inspired by Umemoto et al. [31] who proposed several formulas which include term frequency and their multiplication or division and also various weighting.
3. The more unique the terms, the better distinguishing ability can be obtained between similar or major intentions. Here we have been inspired by [5, 6, 8] who proposed formulas that include term frequency and inverse document frequency.

We found these criteria by experimenting with Equation (3) using data from our preliminary experiments. A more detailed description of them can be found in a later subsection of this paper. The overview of the method is shown in Figure 1.

3.1 AQueReBET Method

Our method is fully automated and thus requires data only from the seeker's gaze within the first SERP and from the seeker's input query. To clearly interpret the process we need to represent the data in a usable form. We decided to use a table τ_x to represent text x (x could be a query, a snippet, or a web page) by a set of pairs consisting of a word w and its importance $im_x(w)$ (we start with word importance within the text x but in further steps we will recalculate it to a wider context):

$$\tau_x = \{(w_i, im_x(w_i) \mid i \in 1, 2, \ldots, n\} = \begin{bmatrix} w_1 & w_2 & \ldots & w_n \\ im_x(w_1) & im_x(w_2) & \ldots & im_x(w_n) \end{bmatrix} \quad (1)$$

where $n = |\tau_x|$ is the number of different words chosen from text x, w_i is the i^{th} word, and $im_x(w_i)$ is the importance of $w_i \mid i \in \{1, 2, \ldots, n\}$ in text x. Initially (before processing the data) all unique words are chosen from text x and the importance of each word is equal to its multiplicity (term frequency) in text x: $im_x(w_i) = tf_x(w_i) \mid i \in \{1, 2, \ldots, n\}$.

The method consists of 6 steps shown in Figure 1. In Step 1, keywords from the initial web search query are selected and the initial table τ_{in} is created:

$$\tau_{in} = \begin{bmatrix} w_1 & w_2 & \ldots & w_n \\ 1 & 1 & \ldots & 1 \end{bmatrix}. \quad (2)$$

For example, if the query is "monk" (e.g. with the aim to find more information about monks in a monastery and their ascetic lives), then

$$\tau_{in} = \{(\text{monk}, 1.0)\} = \begin{bmatrix} \text{monk} \\ 1 \end{bmatrix}.$$

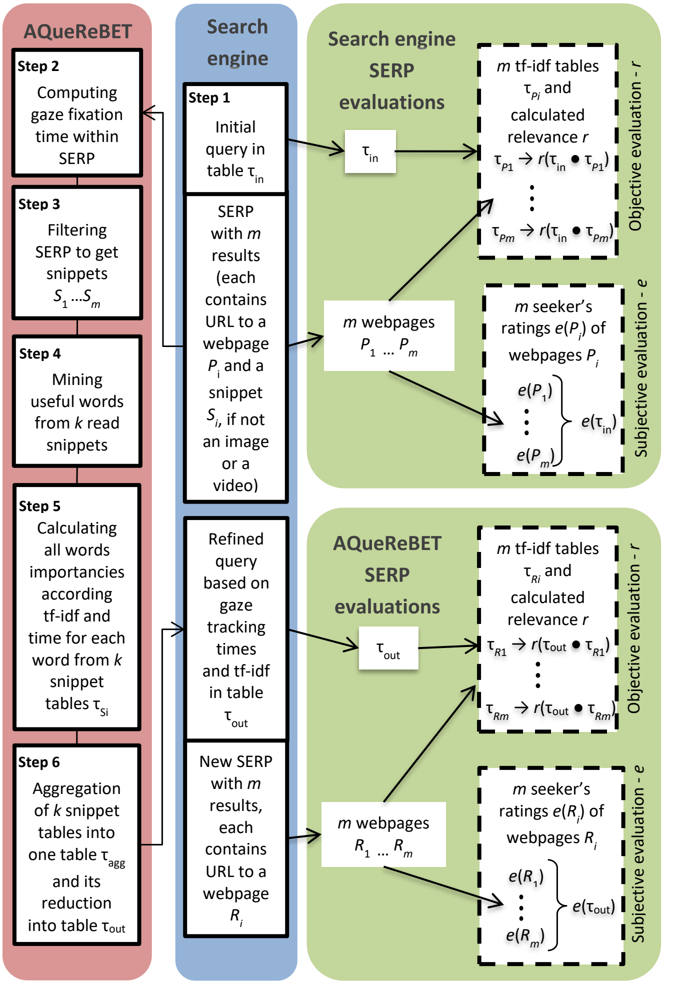


Figure 1. Process of initial query refinement (with new snippets suggested) within AQueReBET and system of its objective and subjective evaluation both search results – with and without AQueReBET

Step 2: *Web seeker's gaze acquisition.* Web search is an interactive experience, typically implemented using dynamic web technologies (using Ajax, DOM rewriting), which is very hard to track accurately using off-the-shelf eye tracking analysis software. To obtain a seeker's gaze in a dynamic web environment, we employ the data collection infrastructure [24] developed at our University User eXperience Research Centre. Raw data from the eye tracker is processed into normalized coordinates of eye position and analysed based on the underlying web page (see Step 2 in Figure 1). The data is subsequently sent to a browser plugin and enriched with XPath data.

Step 3: *Filtering out unnecessary data from SERP to get snippets.* Our data are from SERP within the google.com domain and subdomains (an example of SERP for "monk" is shown in Figure 2).

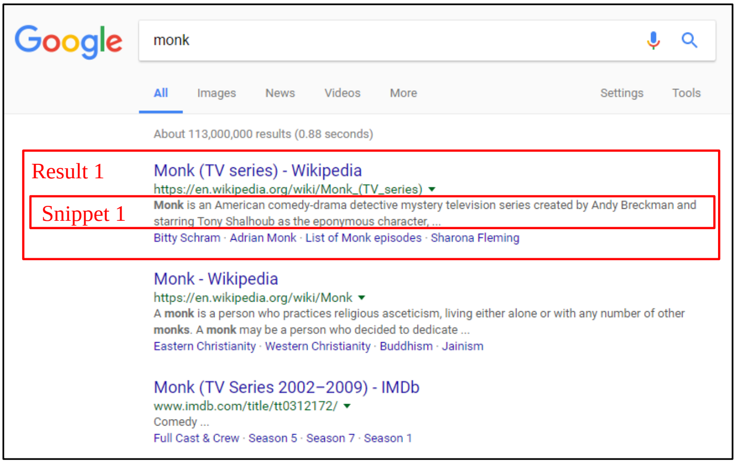


Figure 2. Google's SERP for query "monk", only first three results (snippets) shown

Since we are interested only in snippets and these pages contain also other data, we need to find these snippets within SERP, and to record only snippet areas. From the implementation point of view, the most important part of this process is to filter out the data that do not carry snippets. We used HTML Document Object Model elements (DOM elements), which enabled us to manipulate with HTML elements in DOM. By analysing DOM elements in Google's SERP the page is divided into several sections that contain or do not contain keywords. In this regard it is appropriate to filter out both too large DOM elements, which involve a number of snippets and too small elements that do not carry any relevant information. As a result of this filtration we obtained only the data from all the SERP snippets $S_1, \ldots, S_m$ (see Step 3 in Figure 1, these m snippets are linked to web pages $P_1, \ldots, P_m$). The individual data records we get from the first step contain XPath information, from which we could obtain the element and its value (the snippet text). The seeker's gaze data (the snippets texts) are processed in the following steps in order to obtain additional words on the seeker's intention (also called intent or interest), further refining the initial seeker's query.

Step 4: *Mining the relevant words from relevant snippets and creating their tables.* After that, we filter out from all m snippets only the k relevant ones, $k \leq m$ (see Step 4 in Figure 1) – relevant are those which are gaze inspected, i.e. have at least one seeker's fixation (in eye tracker fixation I-VT filter we set the velocity threshold to 30 degrees/second; this gave us the seekers' minimum fixation time in the interval 200–500 ms). Then we filter out all irrelevant words from all gaze-inspected snippets $S_1, \ldots, S_k$ – we remove all stop words using a modified Wordnet dictionary and perform lemmatization of words (nouns and adjectives) to receive a base form (lemma) of the corresponding word. This bears much of the meaning of the word, contrary to stemming. The importance of irrelevant words is set to 0 and are left out of table representation. An example of the gaze inspected snippet S in SERP is e.g. the second snippet: *"A monk is a person who practices religious asceticism, living either alone or with any number of other monks. A monk may be a person who dedicate ... "*. Then after performing the above mentioned operations, the set of relevant snippet words is table τ_S. The importance of each word w_i is equal to its multiplicity (term frequency) $tf_S(w_i)$ within the processed snipped S.

$$\tau_S = \begin{bmatrix} \text{monk} & \text{person} & \text{religious} & \text{asceticism} & \text{living} & \text{number} & \text{monks} \\ 2 & 2 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}.$$

Step 5: *Performing tf-idf analysis of the snippets.* Having a set of snippets, we need to find words which are specific (unique) in them. If a seeker looks at a particular snippet, the specific word that it contains likely contributes to the expression of the seeker's intention. Those specific words are obtained by applying the td-idf formula [14]. Therefore, in this step we perform tf-idf for all the relevant snippet tables $S_1, \ldots, S_k$ from the previous step (see Step 5 in Figure 1). Its result is tf-idf value for each word in each snippet set: $tfidf_S(w_i)$. This will help us to recognize the relevance of words from gaze inspected snippets $S_1, \ldots, S_k$ compared to all snippets in SERP $S_1, \ldots, S_m$. But the importance of the words should not be determined only by their multiplicity $tf_S(w_i)$ as the main criterion and tf-idf value $tfidf_S(w_i)$ as the second criterion, but also by the relative time the seeker spent on the snippets with their gaze t_S (relative to the averaged time spent on one snippet within SERP). Finally, the importance of each word from table τ_S is given by a combination of three contributing factors: the user's total fixation time within the snippet area (see Figure 2), term frequency in watched snippets and the level of term uniqueness, and calculated by Equation (3).

$$im_S(w_i) = tf_S(w_i) * (tfidf_S(w_i) + t_S) \tag{3}$$

where

$$tfidf_S(w_i) = tf_S(w_i) * idf_S(w_i) = tf_S(w_i) * \log \frac{|\{S_j \in \text{SERP}\}|}{|\{S_j \in \text{SERP} : w_j \in \tau_{S_j}\}|}.$$

Let us suppose, in our example, the dwelling time was 1.2 seconds, averaged dwelling time for all snippets in SERP was 1 second, thus $t_S = 1.2\,\text{s}/1\,\text{s} = 1.2$. Then our example of table τ_S has changed importance values as follows:

$$\tau_S = \begin{bmatrix} \text{monk} & \text{person} & \text{religious} & \text{asceticism} & \text{living} & \text{number} & \text{monks} \\ 2*(0.2+1.2) & 2*(0.22+1.2) & 1*(0.11+1.2) & 1*(0.02+1.2) & 1*(0.06+1.2) & 1*(0.03+1.2) & 1*(0.3+1.2) \end{bmatrix}.$$

And the resulting values of word importance for the snippet S are:

$$\tau_S = \begin{bmatrix} \text{monk} & \text{person} & \text{religious} & \text{asceticism} & \text{living} & \text{number} & \text{monks} \\ 2.8 & 2.84 & 1.31 & 1.22 & 1.26 & 1.23 & 1.5 \end{bmatrix}.$$

Our example demonstrates the calculation of word importance only for one snippet, but the seeker can gaze on more of them: $S_1, \ldots, S_k$. Therefore, the final step is to aggregate tables $\tau_{S_1}, \ldots, \tau_{S_k}$ into table τ_{agg}, which contains the union of all words from tables $\tau_{S_1}, \ldots, \tau_{S_k}$ and their importance:

$$im_{agg}(w_i) = \sum_{j=1}^{k} im_{S_j}(w_i). \tag{4}$$

If a word w_i is not in table τ_{S_j} then $im_{S_j}(w_i) = 0$.

Step 6: *New query definition.* The next step is SERP enrichment with snippets (and corresponding links to pages) that are not present in the original SERP and contain more relevant pages/documents/information sources. Based on the final table τ_{agg} (see Step 6 in Figure 1), we start a new search (as a background process). Since table τ_{agg} can consist of many words, only the most important ones have to be chosen (too many words in a query to a search engine are counterproductive). For example, the new query would be set to the four most important words from table τ_{agg}, resulting to τ_{out}:

$$\tau_{out} = \begin{bmatrix} \text{person} & \text{monk} & \text{monks} & \text{religious} \\ 2.84 & 2.8 & 1.5 & 1.31 \end{bmatrix}.$$

Although the table τ_{out} is reduced to four pairs, it is still a refinement of the initial query represented by table τ_{in}, since it is often shorter. We determined the number of pairs empirically simply by using different queries and four worked out the best. By performing a new search using this new query, we get new SERP with new snippets with links to web pages $R_1, \ldots, R_m$.

3.2 Groupization Method

In general, a groupization is one of the several methods used to improve a personalized web search [16, 28] defined as *"combining an individual's data with that of other related people to enhance the performance of personalized search"*.

In our work the groupization is used in a very specific way. Its purpose is to enhance our AQueReBET method in its last step (see Step 6 in Figure 1). It is performed by using various methods, but still includes the index of similarity. Groups of seekers are formed by two techniques that are based on:

- syntactic similarities with the initial query from the table τ_{in}.
- syntactic similarities with a refined query from the table τ_{out}.

The use of groupization is particularly important due to the fact that a seeker can search for a completely different thing than the one that was written in the initial query. Because of this fact, refinement of a seeker's query using only information elicited from eye tracking data may in some cases have only a limited effect. By also involving the groupization we attempt to remedy such situations. To involve groupization essentially amounts to involving some data from previous experience. In the context of our method, we use groupization to provide results that other seekers adopted as relevant for their query before or during a search session.

We identified a need for implementation of two groupization types, where groups are formed based on the initial queries and the refined query.

1. Initial query from table τ_{in}: After calculation of relevance (Equation (5)) for each page P_i,
2. Refined query from table τ_{out}: After creating a new query defined by table τ_{out} and calculation of relevance (Equation (6)) for each page R_i.

The groupization module in AQueReBET writes one or several of the most relevant pages/documents into a database for the initial or the refined query. This creates a group with a certain query and pages/documents that our system evaluated as relevant for it. We insert only distinct records into the database. If some group already exists in the database, we only update the set of relevant pages/documents. In subsequent search sessions, we attempt to assign a querying seeker to a group that is related to his/her initial query and acquired words by semantic similarity. In return, he/she is provided with the most relevant documents from the group that have been gathered in the previous sessions. If none of the existing groups is relevant enough (i.e. the query is not similar to any group or the level of similarity is too low), a new group is created. Hereafter AQueReBET with a turned on groupization module is AQueReBET + G and with a turned off groupization module is AQueReBET.

4 EXPERIMENTS

To perform experiments evaluating our proposed method requires a specific approach, one of the reasons being that we have not found any other similar systems published in a way that would allow an effective comparison. Experiments that we completed so far deal with the evaluation of our method that refines seekers' queries. We conducted two types of evaluations: Automatic evaluation and evaluation by seekers.

4.1 Automatic Objective Evaluation (Relevance Evaluator)

Automatic evaluation of web pages addressed by snippets without involving seeker's gaze. It is quite possible that a snippet S_i (result in SERP) may not accurately reflect the content of its destination page P_i. It is appropriate to analyse each destination page separately. Since we already have the initial seeker's query (table τ_{in}), we can analyze the relevance of individual pages addressed by a snippet from SERP in the context of this query (see Figure 1, objective evaluations r in Results without AQueReBET). The process of evaluation begins very similarly to the one with snippets. We perform tf-idf on the content of each destination page P and get its table τ_P (the time and aggregation is skipped). To compute relevancy r of each page P_i (examples in Table 1) we use the initial table τ_{in} as follows:

$$r(\tau_{in} \cdot \tau_{P_i}) = \sum_{\forall w \in \tau_{in} \cdot \tau_{P_i}} im_{P_i}(w) \tag{5}$$

where $\tau_{in} \cdot \tau_{P_i}$ represents a set of words, which belong to both τ_{in} and τ_{P_i} and $im_{P_i}(w)$ is the importance of word w from tf-idf analysis on page P_i.

Original documents P_i	norm. rel.
`https://en.wikipedia.org/wiki/Monk_(TV_series)`	0.03
`http://en.wikipedia.org/wiki/Monk`	1
`http://www.imdb.com/title/tt0312172/`	0
`http://www.tv.com/shows/monk/`	0
`http://www2.usanetwork.com/series/monk/`	0
`http://www.newadvent.org/cathen/10487b.htm`	0.48
`https://www.facebook.com/monk`	0
`http://us.battle.net/d3/en/class/monk/`	0
`http://www.battle.net/wow/game/class/monk`	0
Suggested documents R_i	**norm. rel.**
`http://en.wikipedia.org/wiki/Asceticism`	0.24
`http://en.wikipedia.org/wiki/Monk`	0.24
`http://dictionary.reference.com/browse/ascetic`	0.29
`http://www.vocabulary.com/dictionary/ascetic`	1
`http://stgeorgegreenville.org/OurFaith/Articles/Asceticism-Rossi.html`	0.06
`http://www.huffingtonpost.com/...`(very long URL)	0.06
`http://www.monasteryofstjohn.org/...`(very long URL)	0
`http://orthodox.cn/patristics/300sayings_en.htm`	0.02
`http://www.cgg.org/...`(very long URL)	0.17
`https://books.google.sk/...`(very long URL)	0.25

Table 1. Example URLs of web pages P_i and R_i with their normalised relevance (Equation (7)). Normalized relevance computation does not involve seeker's gaze.

The second document P_2 has the highest normalised relevancy because it is the closest to the seeker's intention from among the original set of documents P_i. There is another document with a quite high normalised relevance of 0.48 and indeed, it at least partially fits the seeker's intention, too. The remaining documents deal with entirely different things (e.g., a TV series, a computer game).

Automatic evaluation of new web pages addressed by new snippets. To measure relevancy of m new pages/documents R_i (examples in Table 1), we perform a new tf-idf analysis of these pages in the context of the new query from the table τ_{out} (see Figure 1, objective evaluations r in Results from AQueReBET). For computing the table τ_{R_i} we determine the tf-idf importance of individual words on page R_i. We use the same process as for pages P_i: We download the content of pages $R_1, \ldots, R_m$, then calculate the tf-idf analysis of these pages to get tables $\tau_{R_1}, \ldots, \tau_{R_m}$, compute $\tau_{out} \cdot \tau_{R_i}$ and finally we sum up the importance of queried words for each page R_i to get its relevance $r(\tau_{out} \cdot \tau_{R_i})$:

$$r(\tau_{out} \cdot \tau_{R_i}) = \sum_{\forall w \in \tau_{out} \cdot \tau_{R_i}} im_{R_i}(w). \tag{6}$$

Normalisation. For each relevance value of page P_i (alternatively page R_i from a new SERP) we also perform normalisation into interval $[0, 1]$ using feature scaling:

$$||r(\tau_{in} \cdot \tau_{P_i})|| = \frac{r(\tau_{in} \cdot \tau_{P_i}) - \min_{j \in \{1,\ldots,m\}} r(\tau_{in} \cdot \tau_{P_j})}{\max_{j \in \{1,\ldots,m\}} r(\tau_{in} \cdot \tau_{P_j}) - \min_{j \in \{1,\ldots,m\}} r(\tau_{in} \cdot \tau_{P_j})}. \tag{7}$$

Examples of calculated normalised relevance are in Table 1 for both P_i and R_i. A page with normalised relevance equal to 1 represents the most relevant document and a page with 0 represents the least relevant document. We introduce normalisation under the assumption that search results are at least partially different. In the very unlikely case that all the results in SERP have the same relevance, the normalisation would not work. On the other hand, we need to normalise, because we need to compare:

- First, after normalisation we can compare the relevance of different queries. This would be otherwise difficult, since different queries may generate widely differing relevance, e.g. in one case in range 0–1 000, in another case 0–5. To compare them without normalisation would be misleading. Here, the fact that the pages' relevance values are distributed similarly for any query is very helpful.
- Second, we need to normalise relevance to be able to compare it with seekers' evaluations (each seeker ranks both P_i and R_i pages – for more details see the next section). This comparison ensures that we calculated relevance correctly. Correctly calculated relevance can be used to extend the AQueReBET method, e.g. by automatic SERPs browsing (the next SERPs for a set

query) and filter out only the pages with the best relevance, order them and offer them to a seeker.

- Third, the above-mentioned normalisation helps other researchers to compare their results with ours.

4.2 Subjective Evaluation by Seekers

Eye tracker settings. The experiment was conducted in the User eXperience Research Centre at the Faculty of Informatics and Information Technologies, Slovak University of Technology in Bratislava. The Centre consists of two eye-tracking-enabled laboratories. The data collection was performed in the laboratory for detailed research of user experience using a Tobii TX300 eye tracker. The Tobii TX300 is a high precision remote eye tracker with average accuracy 0.4 deg (under ideal conditions), processing latency 1.0 to 3.3 ms, total system latency at most 10 ms, and blink recovery time 10 to 165 ms. The eye tracker is able to compensate for large head movements enabling unobtrusive research. Gaze data was acquired in binocular mode at 300 Hz sampling rate. Participants took part in the study separately. Each participant was comfortably seated at 65 cm eye distance from the eye tracker, calibrated and verified using the live viewer.

Participants. 22 participants took part in this second evaluation. All of them were university students, 6 females and 16 males. Each participant received five queries with specific search intent, and subsequently evaluated (through explicit ratings) the relevance of documents that were obtained with and without using our method. Following that, we tested the impact of using the proposed groupization approach on respondent's satisfaction with documents' relevance. The last step of the experiments was to compare calculated documents' relevance produced by our system versus evaluation of the same documents' assessed by the respondents. We divided our participants into two equally-sized groups. Both groups consisted of 11 participants. All of the participants queried Google with the given intentions. Participants in the first group received results from Google and then from AQueReBET, participants in the second group received results from Google and then from the AQueReBET with the groupization on. Similarly, those 22 participants were divided into 2 groups based on their IT skills: IT-participants with higher search skills (mostly students of information technology related study programs), and non-IT-participants with lower web search skills. Both groups consisted of 11 participants. In the IT group 5 participants had turned on the groupization in AQueReBET and 6 off; in the non-IT group, 6 on and 5 off.

Experiment settings. We allocated a time slot of 30–40 minutes for each of the 22 participants (seekers). The time slot includes eye tracker calibration, an explanation of experiments and queries, and answering seeker's questions. Seekers searched answers for the queries from Table 2 during their sessions with specific search intent with the aim to find the most relevant pages/documents for the particular query. Seekers were allowed to set a given search query into a search bar of `www.google.com` search engine in a private window of the Google Chrome browser and look through the search results.

Query ID	Initial Query	Complexity (Level)	Interest/Intent
Q1	Monk	Understand (2)	Find more information about monks in monasteries and their ascetic lives.
Q2	Major	Remember (1)	What is the meaning of major in music in the context of the musical scale?
Q3	Hockey stick	Evaluate (4)	Where to buy a hockey stick?
Q4	Amnesiac band*	Analyze (3)	Retrieve information about Radiohead's album Amnesiac. However, there also exists a musical band called Amnesiac.
Q5	Australian second world war	Analyze (3)	Service records details of Australian soldiers who fought in the Second World War.

Table 2. Queries and interests (* initial query for Q4 is intentionally wrong to find out if AQueReBET can correct it and give the correct pages in SERP)

They first looked at the results from Google and if they were unsatisfied, they could click on any of them and read the new page/document. They could even amend the original query. After a few seconds, based on the data collected from the seeker's gaze, AQueReBET offered them its search results (suggested additional relevant pages/documents) and they were allowed to do with it the same as with the first one from Google. Finally, they rated both sets of results. The order of presenting the Google and the AQueReBET results cannot be changed, since results from AQueReBET depend on results from Google, which have to be seen first. The AQueReBET evaluations have to be done by the same person, who saw the Google SERP first. The fixed order can, of course, shift our results, but since both SERPs contained a mixed order of right and wrong results, this keeps the seeker's evaluations very closely level to the seeker's objectivity. In later results, we consider the seekers' evaluations as not shifted.

Query determination. Queries were selected based on several aspects: query ambiguity, query complexity in context of length, cognitive complexity of intention and its domain [2, 33], and the number of relevant documents retrieved by the initial SERP provided by Google. Most of the participants did not have any prior knowledge about any query subject. In a few cases, they did have some prior knowledge but we found their prior knowledge was unrelated to the evaluated scenario and we asked them to ignore it to reduce the bias as much as possible. Q4 is a special query, using which we wanted to evaluate the case with a partly wrong initial query ("Amnesiac band" instead of "album Amnesiac" –

see Table 2) but using the their gaze data, the system was able to correct it and provide the relevant output.

Questionnaires (seekers' subjective evaluations). Seekers subsequently evaluated through explicit ratings documents retrieved by the initial SERP (provided by Google in response to the initial query) and those retrieved by our enhanced SERP (provided by Google in response to a query refined by AQueReBET), and after that they answered a questionnaire. In this questionnaire they filled the suggested relevance score e as a value from 0 (absolutely irrelevant) to 10 (absolutely relevant) for each page/document (see Figure 1: subjective evaluations – e in Results without AQueReBET and in Results from AQueReBET). They also indicated an overall satisfaction rate of the suggested documents for each query. For results provided by Google in response to a query refined by AQueReBET, we shall use the formulation "provided by AQueReBET." We wish to emphasize that the role of the Google search engine is twofold. It was quite natural to use it as the "base" search engine upon which to build our extension. But despite our efforts to find results of similar research that could be used for a direct comparison, we found none and therefore have chosen Google also as a system to compare with – of course within the limited scope of our goals.

Data collection reliability. Eye tracking data collection can be unreliable when used with consumer or lower precision research eye trackers, or when the experimental setup is not congruent with the eye tracker's capabilities. The size and spacing of AOIs (areas of interest) on which the metrics are computed need to be large enough so that the gaze points are not misattributed to a wrong AOI. In our case, we used a very robust experimental design:

1. we used relatively large areas of interest – the whole snippets in results page,
2. we employed simple gaze metric (total fixation duration), and also
3. we used a high precision eye tracker (Tobii TX300) that is a time proven robust device that would be able to provide reliable measurements even on word level, or saccadic movements which we did not analyze.

Data quality for each participant was verified by the experiment moderator.

4.3 Metrics

There are many different metrics used to measure the success or effectiveness of information retrieval (through explicit seeker's ratings). We choose the following as they fit our aim (method evaluation) and they are also used by other authors, what allows us to compare to them.

The discounted cumulative gain (DCG), was introduced by [12]. They explained that DCG reflects the fact that "the greater the ranked position of a relevant document, the less valuable it is for the user, because the less likely it is that the user will ever examine the document due to time, effort, and cumulated information from documents already seen." Moreover, they introduce normalised discounted cumulative gain (nDCG), to be able to compare different DCG curves (e.g. those that use different ranges of seekers' evaluations). These metrics are determined using the following formulas:

$$DCG@k(Q_j) = \sum_{i=1}^{k} \frac{2^{||e(R_i)||} - 1}{\log_2(i+1)} \quad \text{and} \quad nDCG@k(Q_j) = \frac{DCG@k(Q_j)}{\sum_{i=1}^{k} \frac{1}{\log_2(i+1)}} \tag{8}$$

where $||e(R_i)||$ represents a normalised satisfaction rate of a page/document R_i (either provided by Google or by AQueReBET). Our seekers rate pages using an 11 point Likert's scale – $e(R_i) \in \{0, 1, 2, \ldots, 10\}$. Ratings are afterwards normalised (divided by 10) into the interval $[0, 1]$, where 0 means no satisfaction with the document and relevance 1 means the maximum satisfaction with the document relevance. The number k represents the number the first k results in SERP for one set query Q_j. The denominator in nDCG metrics represents the norm – the ideal performance, where all the evaluations are equal to 1. In later evaluations we use also averaged nDCG@k:

$$DCG@k = \frac{1}{|Q|} \sum_{j=1}^{|Q|} DCG@k(Q_j) \quad \text{and} \quad nDCG@k = \frac{1}{|Q|} \sum_{j=1}^{|Q|} nDCG@k(Q_j) \tag{9}$$

where $|Q|$ is the number of different queries.

We also calculated mean average precision (MAP), which is similar to nDCG, since both have a maximum equal to 1 and both decrease with each irrelevant result in SERP. Unlike nDCG, MAP uses only binary classification (0 for irrelevant result and 1 for relevant result) and for a set of queries is the mean of the average precision scores for each query [31, 23, 5]:

$$MAP@k(Q_j) = \frac{1}{k} \sum_{i=1}^{k} \text{Prec}@i(Q_j) \text{ and } MAP@k = \frac{1}{|Q|} \sum_{j=1}^{|Q|} MAP@k(Q_j) \tag{10}$$

where $|Q|$ is the number of different queries, Prec@$i(Q_j)$ is computed as the fraction of relevant documents within the top i results for query Q_j. Since our seekers did not use binary classification, we had to set the threshold from which the results are relevant and the rest had to be irrelevant. During our pilot tests we noticed that seekers used the following criteria to divide the scale in 5 parts:

- the first (values equal to 10) – exactly what was seeker looking for,
- the second (values 7, 8, 9) for relevant results,
- the middle part (values 4, 5, 6) for partly relevant results,
- the fourth (values 1, 2, 3) for irrelevant results and

- finally the last one (values equal to 0) for totally irrelevant result.

This division is noticeable in our histograms (see Figure 3), where one of the local maximums is usually at 7 or 8 and the other at 4 or 5. This division is similar also to other authors (e.g. [27]). Based on other studies (including [5] we decided to use the border value number 4, since from this value there are at least partly relevant documents. It means if a seeker evaluated a result by $e \geq 4$, we calculated with it as with a relevant result (in binary classification precision equal to $1 =$ positive) and if the evaluation $e < 4$, then we calculated with it as with an irrelevant result (in binary classification precision equal to $0 =$ negative). Another metric used to evaluate a query is the reformulation necessity called $Rfactor$. It takes into account the number of times that a query had to be reformulated on average. It is determined using the following formula:

$$\text{Rfactor} = \frac{\sum_{i=1}^{|Q|} \text{Refor}(Q_i)}{|Q|} \tag{11}$$

where $\text{Refor}(Q_i)$ is the number of necessary reformulations of initial query Q_i until the relevant results are achieved. If the query Q_i has not been reformulated by the seeker even once, $Refor(Q_i) = 0$. $|Q|$ represents the number of initial queries (in our case, $|Q| = 5$).

Satisfaction with the whole result generated by the initial or a refined query evaluated (= rated) by i^{th} seeker is $e(\tau_{int})$ or $e(\tau_{out})$, respectively. It was a subjective evaluation, where seekers used the same Likert's scale afterwards normalised to interval $[0, 1]$. This number should have been in correlation with nDCG.

4.4 Comparison of Automatic Evaluation and Evaluation by Seekers

To compare our automatic evaluation with the evaluation done by seekers, we used standard metrics: $accuracy = \frac{TP+TN}{P+N}$, $recall = \frac{TP}{TP+FN}$, $precision = \frac{TP}{TP+FP}$, and $F = 2 * \frac{precision * recall}{precision + recall}$, where TP or "true positive" is the number of correctly classified relevant documents, TN or "true negative" is the number of correctly classified irrelevant documents, P represents the number of relevant documents and N the number of irrelevant documents, FN or "false negative" is the number of incorrectly classified relevant documents and FP or "false positive" represents the number of incorrectly classified irrelevant documents. In the context of a web search, these metrics are calculated from the displayed results on SERP (mostly 8, 9 or 10 of them). To calculate these metrics, we used our calculated normalised relevance $||r|| \in [0, 1]$ and seekers' evaluations $e \in \{0, 1, \ldots, 10\}$. To compute these metrics we need to convert both in binary classification. As we already explained, we set for seekers' evaluations the borderline number 4. For relevance $||r||$ it was a bit complicated, so we calculated the trend line between e and $||r||$ from our pilot tests. It came out that border $||r|| = 0.1$ corresponds to border $e = 4$. It means that if AQueReBET relevance value $||r|| < 0.1$, the result is negative, if $||r|| \geq 0.1$, positive.

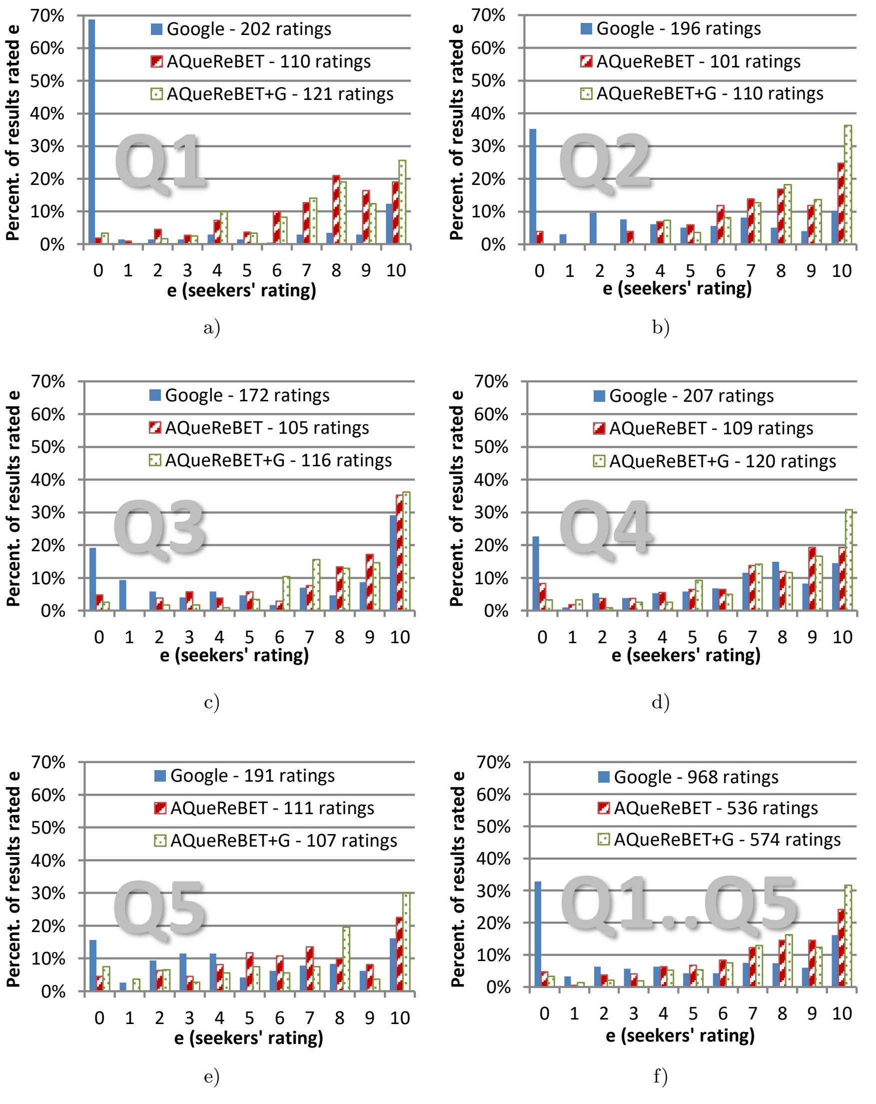


Figure 3. Normalised distribution (histogram) of seekers' ratings (how they rated results from SERP) for query a) Q1, b) Q2, c) Q3, d) Q4, e) Q5 and f) all five queries Q1, Q2, Q3, Q4, Q5 together. Comparison between results obtained using the Google engine (blue columns), our AQueReBET system (red columns) and our system enhanced by groupization – AQueReBET + G (green columns). All ratings e appear on axis x from $e = 0$ – irrelevant information source, to $e = 10$ – exactly the information source I wanted.

It should be noted that we do not present any correlation between ratings by seekers and AQueReBET. When attempting to compute it, we realised that the statistical distributions of the respective ratings are very different (cf. Table 1 and Figure 3) and therefore it is not appropriate to calculate the correlations.

5 RESULTS OF EXPERIMENTS

In this section, we describe the results of some of our experiments. We try to compare the achieved results with those achieved by other authors. This is not entirely possible due to different sources of data and different outputs of the corresponding methods. Therefore, any claim we shall make regarding superiority of any of those approaches is of limited validity only. In particular, when results of our method are better than results of another method, it may be due to the method itself or due to the experiment set up and we are not able to tell which is the case.

5.1 Google as a Baseline Versus AQueReBET Evaluated by Seekers

The primary goal of the performed experiments is to falsify or endorse the hypothesis that when we have the data from a seeker's gaze, we can suggest more relevant documents to the seeker and reduce the need for reformulation of the initial query.

To find out, we let all the seekers rate relevance of each document (see Figure 1 subjective evaluations – e), either provided by Google, by AQueReBET or by AQueReBET + G. At first, we compared these three response providers using the DCG@k, nDCG@k metrics for k from 1 to 9 (Figures 4 c) and 4 d)); to show the difference between individual queries we added also DCG@5(Qi) and nDCG@5(Qi) (Figures 4 a) and 4 b)).

Graphs in Figures 4 a) and 4 b) show that values of measures DCG and nDCG depend quite strongly on the type of query. Generally, however, we observe that our answer provider AQueReBET gives better results as the baseline (Google). In some cases, groupization is able to yield further, albeit slight improvement. Graphs in Figures 4 c) and 4 d) show how values of DCG and nDCG change with an increasing k. We note that the DCG measure in Figure 4 c) is somewhat harder to read due to the fact that the ideal maximum value changes with k. The nDCG measure of the baseline gives the smallest value for $k = 1$. This is caused by the ambiguity of queries, which are too short. In case of AQueReBET, all queries have been refined, so there is not such a steep increase of values from $k = 1$ to $k = 2$. Both AQueReBET curves show statistically significant improvement over the baseline for every k ($p < 0.05$, Wilcoxon signed-rank test). For example, for $k = 5$ we have a significant improvement from 38 % ± 17 % (Google) to 74 % ± 13 % (AQueReBET) alternatively 78 % ± 14 % (AQueReBET + G) representing a rise by 36 alternatively 40 percentage points as well as 1.9 alternatively 2.1 times increase.

When comparing nDCG@5 of SERPs provided by AQueReBET and AQueReBET + G for the whole dataset, groupization improved it by only approximately

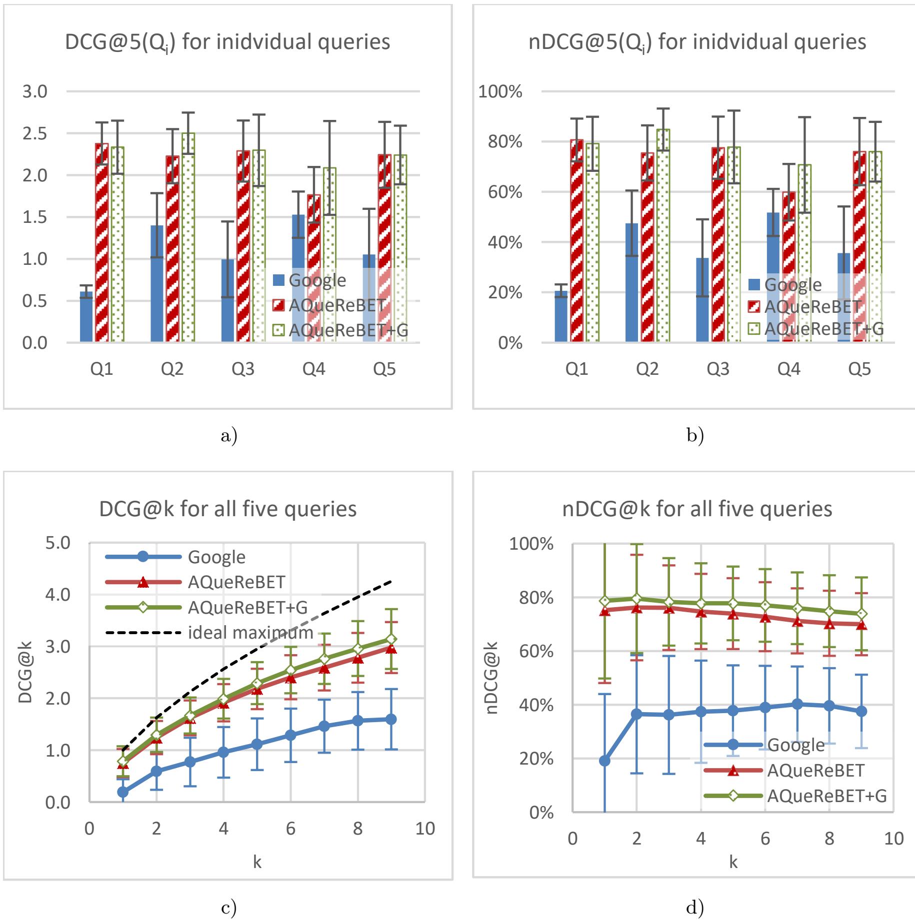


Figure 4. a) DCG@5 for individual queries, b) nDCG@5 for individual queries, c) DCG@k for all five queries and d) nDCG@k for all five queries averages with standard deviations when Google, AQueReBET, AQueReBET + G is used

4 percentage points. However, the difference is not a statistically significant improvement (Mann-Whitney Test for Two Independent Samples, with $\alpha = 0.05$, two tailed, p-value $= 0.11$ did not refute the hypotheses about the same means). This means that most of the times the refined query (and consequently its SERP) was already good enough without using groupization. In such circumstances, the groupization cannot bring tangible improvements. We hypothesize that this value should increase more significantly by enlarging the groupization database over time.

The difference in the individual averages of DCG@5 and nDCG@5 (see Figures 4 a) and 4 b)) shows that the results for individual queries are diverse and thus should be evaluated separately, too. Their diversity is also visible in Figures 3 a), 3 b), 3 c), 3 d) and 3 e), minor differences between the number of ratings are due to the exclusion of invalid results, e.g. lack of rates, and so on. Nevertheless, we found that AQueReBET, by enriching the web search by data collected from seekers' gaze, improves the relevance of documents retrieved significantly ($p < 0.05$) except the results for Q4.

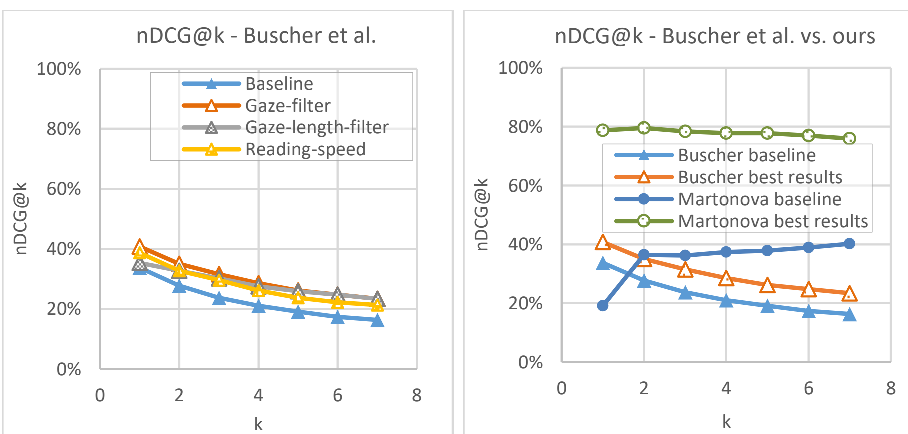
Let us discuss all the queries one by one: The reason for such considerable DCG@5 and nDCG@5 improvement for Q1 and Q2 dwells probably in their ambiguity. These queries have more than one interpretation and therefore Google itself was not able to provide SERP with pages, which fit to the meaning the seeker had in mind (see the highest first column for Google in Figures 3 a) and 3 b)). Similarly, there is significant improvement for Q3 and Q5 because it is hard to guess from the set keywords what the user's exact intention is. Q5 is definitely the most complex query, therefore it is interesting how the averaged nDCG@5(Q5) increased from $36\,\% \pm 16\,\%$ to $76\,\% \pm 13\,\%$ (for more details see Figure 3 e), Figure 4 b)). Q4 did not pass the border value $\alpha = 0.05$ probably because of the low number of pairs in a sample and also the specific type of query – it is the one with intentionally partly wrong input. Its averaged nDCG@5(Q4) of the initial SERP (provided by Google) is $53\,\% \pm 10\,\%$ (sample1) respectively $51\,\% \pm 9\,\%$ (sample3), what seems high for a partly incorrect query, but AQueReBET provided SERP with an even higher $60\,\% \pm 11\,\%$ (statistically insignificant increase, $p = 0.11$) and $71\,\% \pm 19\,\%$ (when groupization involved, statistically significant increase, $p < 0.05$, see Figure 4 a), for more details see Figure 3 d)).

When interpreting these results, one should keep in mind three facts:

- The Google result page for the same query may slightly vary, since the experiment was performed over several days.
- When different seekers input the same initial query to Google, it may lead to different refined queries for AQueReBET, since individual seekers' gaze patterns can lead to different tables (with words and their importance) for each seeker. Because of the same reason the suggested pages and documents on the AQueReBET result page vary for different seekers and even if there are some the same, they have different relevance.
- Two different seekers rate the same SERP slightly differently (subjective evaluation). This is obviously more likely in the case of a greater number of seekers. Thus the maximal achievable nDCG (or any other metrics) for any query shall most likely not be $100\,\%$. We assume the maximal achievable nDCG is in the top decile.

It would be beneficial, if not compulsory, to compare our results with some representative works of others who have dealt with a similar problem. For our work, it is instructive to make a comparison, e.g. with the influential work of [5]. It should be noted, however, that there were differences in experiments. Contrary to ours, their users browsed whole documents (our users read snippets only). Also, the users' tasks were different.

Since Buscher et al. [5] evaluated the DCG metric based on values from the set of ratings $\{0, 1, 2, 3\}$ whereas we used the set $\{0, 0.1, 0.2, \ldots, 1\}$ it was necessary to perform a normalisation. Thus we at first normalised their DCG values to nDCG (see Figure 5 a)) to be able to compare it with our results. In Figure 5 b) one can see that values for respective baseline cases both stay within the 20–40 % interval. However, there is a difference with respect to k: while theirs decreases, ours increases. We assume that tendency of the nDCG@k measure depends on the kind and ambiguity of the given query. In the case of our baseline, queries were mostly ambiguous. As far as the problem itself is concerned, from some abstract view it is possible to treat both classes of problems, i.e. the one in [5] and ours as essentially comparable. This could be claimed while noting that they actually were solving a slightly different problem with a different data set. They also processed data from the eye tracker in a slightly different way. Still, an elementary comparison is possible. They achieved (for $k = 5$) a 1.37 fold improvement, we achieved a 2.06 fold improvement. Their maximum value of nDCG@k is 40.8 % (for $k = 7$), whereas ours is 79.5 % (for $k = 2$). This is, even respecting the 13.7 % standard deviation, considerably better. [6] performed a very similar experiment achieving DCG@10 = 9.15 which corresponds to nDCG@10 = 29.2 % for a read length of 150 characters. Snippets are about 150 characters long. Our result of nDCG@9 = 70 % makes us 2.4 times better.



a) b)

Figure 5. a) nDCG@k for all variants (calculated from their DCG@k [5] and b) nDCG@k comparison of [5] base line with their best results and our base line with our best results

Eickhoff et at. [8] used Mean Reciprocal Rank (MRR) metric:

$$MRR = \frac{1}{|Q|}\sum_{i=1}^{|Q|}\frac{1}{rank_i} \tag{12}$$

where $rank_i$ is the rank position of the first relevant document for the i^{th} query and Q is the set of evaluated queries) attaining values 0.80 and 0.86, compared with 0.79 MRR value for Buscher et al. [5]. In our method, MRR metric for relevant results (seeker's evaluation value 7 or higher) is 0.87 for AQueReBET and 0.91 for AQueReBET + G.

Umemoto et al. [31] is another related work which used nDCG to evaluate their results. They achieved 81.6 % and our best value of nDCG@2 = 79.5 ± 20.3 %. The difference is probably not statistically significant. The main difference in evaluation is that they considered up to 15 terms, whereas we considered 4 words and their rating scale had 3 degrees, but our had 11 degrees. On the other hand, we received a better MAP metric (more details see below).

We also evaluate Prec@k and MAP@k metrics for k from 1 to 9 (Figures 6 c) and 6 d)); to show the difference between individual queries we added also Prec@5(Qi) and MAP@5(Qi) (Figures 6 a) and 6 b)). In Figures 6 a) and 6 b) one can see that both Prec and MAP measures depend, similarly to Figure 4, on the type of query. The effect of groupization is also similar, i.e. sometimes it may improve, but sometimes it may worsen the results of the AQueReBET. In Figures 6 c) and 6 d), the curve for the baseline case initially exhibits a similar steep increase for similar reasons as before (cf. our comments to Figure 4). All in all, however, all curves in Figure 6 look better in the sense they are closer to 100 %. This is caused by the binary evaluation scale of these metrics. A binary evaluation tends to supress small differences that can be observed in DCG and nDCG metrics (we used an evaluation with 11 different values there). As a consequence, differences between results with and without a groupization are almost invisible and look almost identical. But here, too, one can see that both AQueReBET curves show statistically significant improvement over the baseline for every k ($p < 0.05$). For example, for Prec@5 we have an improvement from 50 % ± 27 % (Google) to 91 % ± 14 % (AQueReBET) alternatively 95 % ± 10 % (AQueReBET + G), which represents a 1.8 alternatively 1.9 times increase. For for MAP@5 we have an improvement from 44 % ± 28 % (Google) to 93 % ± 15 % (AQueReBET and the same for AQueReBET + G), which represents an increase of 2.12 times. When comparing AQueReBET and AQueReBET + G, it is obvious, the difference is statistically insignificant. The results for individual queries (see Figures 6 a) and 6 b)) resemble the results for nDCG, which were already discussed.

In an attempt somehow to compare our results with those of [5], we note that their absolute MAP for baseline is 46.6 % (depending on the used variant from 29.7 % to 54.3 %) and absolute MAP of their method is 55.9 % (depending on the used variant from 39.3 % to 66.7 %), which represents an increase of 1.20 times.

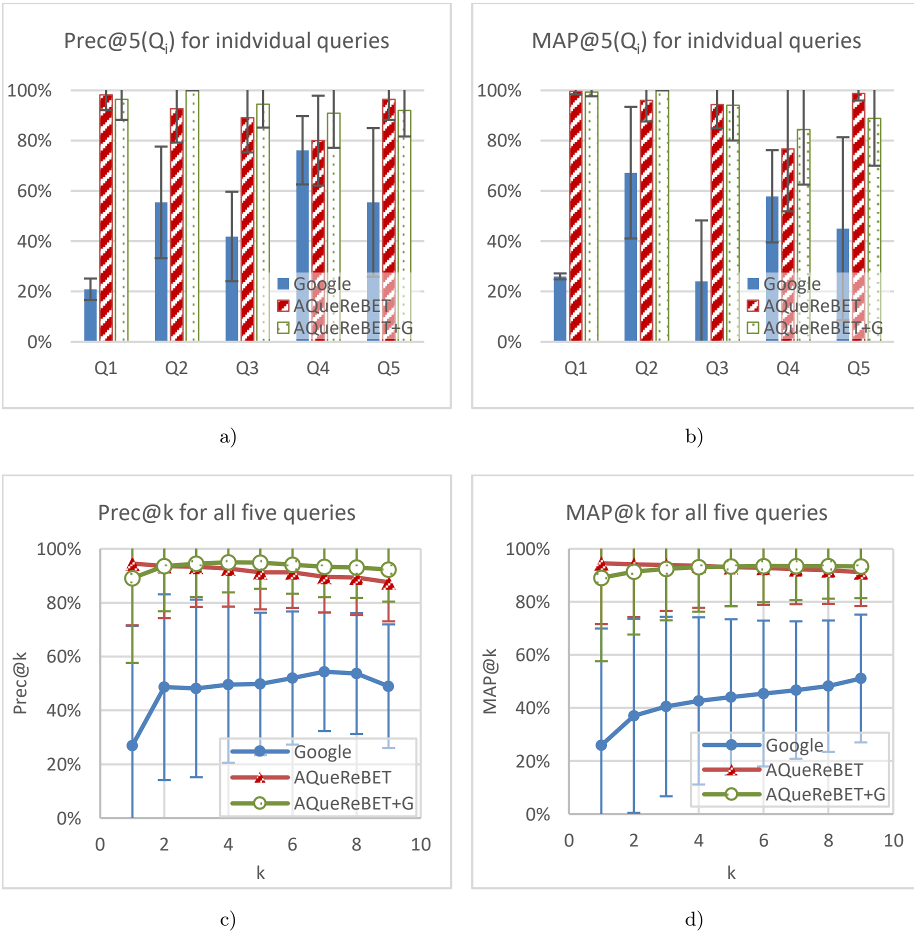


Figure 6. a) Prec@5 for individual queries, b) MAP@5 for individual queries, c) Prec@k for all five queries and d) MAP@k for all five queries averages with standard deviations when Google, AQueReBET, AQueReBET + G is used

Since our and their baseline results are very close, we consider our results considerably better, whether absolute improvement (our 93 % vs. their 55.9 %) or relative improvement (our 2.12 times vs. their 1.20 times increase). [6] experimented in a very similar way achieving MAP@10 = 73.6 % for read length of 150 characters. Snippets are about 150 characters long. Our result of 93 % makes us 1.26 times better. [31] achieved up to MAP = 65.2 % and our best value of MAP@1 = 94.5 ± 22.9 %. However, these results have been achieved under slightly different conditions so they can serve for a rough comparison only. The main difference in evaluation is that they considered up to 15 terms whereas we considered

4 words and their binary mapping of user evaluations uses a different threshold than does ours.

	Rfactor		Satisfaction		Time [s]	
	avg	stdev	avg	stdev	avg	stdev
AQueReBET	25.5 %	9.3 %	76.0 %	4.9 %	66.1	32.6
AQueReBET + G	21.8 %	10.8 %	80.2 %	6.3 %	47.1	13.6

Table 3. Averages and standard deviations of selected metrics for AQueReBET and AQueReBET + G for all five queries together

In Table 3 Rfactor, Satisfaction and time metrics are shown. These are not evaluated for Google, but we can see, that in all of them the AQueReBET + G gives slightly better results than AQueReBET: Rfactor is 3.7 percentage points better, overall satisfaction 4.2 percentage points better and Time 18.7 s shorter.

The AQueReBET average time of presenting the suggested documents was 66.1 (47.6 seconds for AQueReBET + G). Zhu and Mishne [35] presented that the average time for choosing a document as relevant in a web search is approximately 46.15 seconds. That is, provided that the eye tracker calibration is completed, our solution needs only approximately 20 s respectively 1.5 seconds more to show suggested documents. But this strongly depends on calibration, internet connection and seekers' search stereotypes.

5.2 Accuracy of Automatic Evaluation

The power of the AQueReBET system lies not only in the detection of gazed snippets but also in the system's ability to correctly anticipate the seekers' ratings. To evaluate the correctness of the system's calculations we used standard relevance measures – F-measure, precision, recall and accuracy. We calculated them by comparing the seeker's evaluation (rating) e and our system's calculated normalised relevance $||r||$ (see Figure 1 for subjective and objective evaluations).

We should like to note that the former three metrics give higher values with an increasing share of true positives. It is therefore to be expected that results from Google achieve lower values than those acquired by applying our method. On the other hand, the latter accuracy metric reflects only the volume of correctly identified results; it increases with the increasing sum of true positives and true negatives. Provided our automatic evaluation is correct, accuracy shall be the same regardless of the search engine used. This is the reason why, in this part, we present the combined results of all three methods. However, also partial results according to the input query and the combined results for all five queries are included there. We should also note that the achievable maximum for all four metrics can practically never be 100 %, since seekers can differ in their subjective ratings.

Averages with standard deviations of all four relevance metrics for individual queries using different search engines can be found in column graphs in Figure 7.

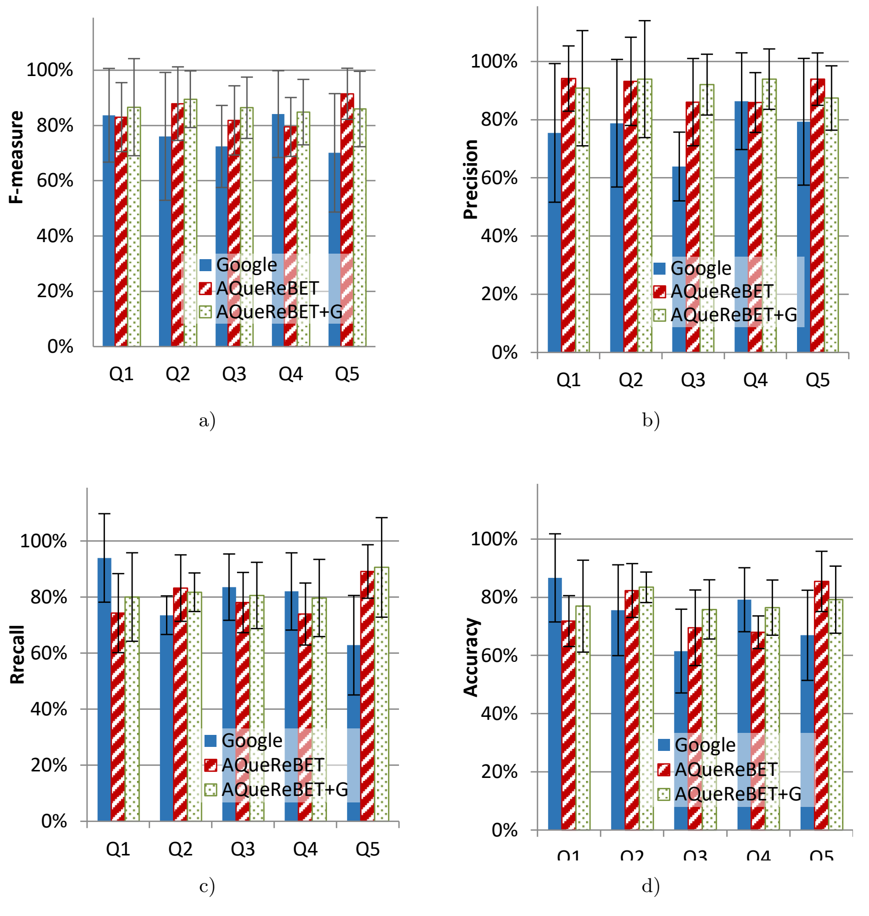


Figure 7. a) F-measure, b) precision, c) recall and d) accuracy averages with standard deviations for queries Q1, Q2, Q3, Q4 and Q5 when Google, AQueReBET, AQueReBET + G is used

When comparing these measures between the individual queries (see Figures 7 a), 7 b), 7 c) and 7 d), we can see that it is not generally the case that the AQueReBET has better averages than Google, or that with groupization it is mostly better. The small difference made by groupization has the same reason as with nDCG – most of the times the refined query (and consequently its SERP) was already good enough without using groupization. When calculated, the statistical significance of differences in averages between Google, AQueReBET and AQueReBET + G, many of them were not significant (F-measure for Q1 and Q4, precision for Q1, Q2 and Q4, recall for Q2, Q3, Q4 and accuracy for Q2 and Q4, same for AQueReBET and AQueReBET + G). It was probably caused by the low number of samples (only 10 or 11 for individual queries, which is not enough for such wide standard deviations and such close averages).

When taking all queries together, F-measure and precision gave us a significant difference ($p < 0.05$) between Google and AQueReBET either with or without groupization. E.g. for F-measure, there was improvement from 71 % ± 17 % (Google) to 84 % ± 10 % (AQueReBET) and from 78 % ± 15 % (Google) to 85 % ± 11 % (AQueReBET + G), which is 13 respectively 7 percentage points improvement. As expected, values for accuracy are nearly the same: 71 % ± 17 % (Google Sample1), 77 % ± 16 % (Google Sample3), 75 % ± 12 % (AQueReBET) and 78 % ± 11 % (AQueReBET + G) – the Wilcoxon test approved that the differences between averages are not significant.

Another point of view gives us results calculated without the search engine differentiation. Values for all search engines together, for individual queries as well as for all of them, are in Table 4 (calculated directly from *TP*, *TN*, *FP*, *FN* counts). The differences between individual queries showed that the queries could give us slightly different results. When summing up all queries together, F-measure, precision, recall and accuracy are in all cases greater than 75 % (see Table 4, grey row), what we consider to be the good quality of our automatic evaluator.

	Rating	TP	TN	FP	FN	F-Measure	Precision	Recall	Accuracy
Q1	431	206	140	35	50	82.9 %	85.5 %	80.5 %	80.3 %
Q2	407	220	103	22	62	84.0 %	90.9 %	78.0 %	79.4 %
Q3	391	241	24	67	59	79.3 %	78.2 %	80.3 %	67.8 %
Q4	435	262	67	36	70	83.2 %	87.9 %	78.9 %	75.6 %
Q5	407	227	78	36	66	81.7 %	86.3 %	77.5 %	74.9 %
$\sum$	2 017	1 156	412	196	307	82.1 %	85.5 %	79.0 %	75.7 %

Table 4. Number of seekers' ratings, number of true positive, true negative, false positive and false negative identifications of our relevancy evaluator, calculated metrics (recall, precision, accuracy and F-measure) for all five queries together and separately as well (not differentiating a type of used search engine)

5.3 Comparison of IT and NonIT Seekers

In the last experiment we compared the results for IT and nonIT seekers' groups. Figure 8 depicts the comparison of nDCG obtained by Google, AQueReBET and AQueReBET + G. As an interesting fact, it appears that groupization significantly improves relevance of documents/pages mainly for the nonIT group (see Figure 8 queries Q1, Q2, Q3 and partly Q4). The results indicate that groupization is helpful especially for seekers who have lower web searching skills. It seems that it is precisely the type of skills that are conveyed to those seekers by the groupization. We also hypothesize that groupization is most effective especially in cases when intentions behind queries are similar across the group. The statistical significance of the difference between IT and nonIT groups was not evaluated since the groups had only 5 and 6 participants and moreover, an improvement was not observed for all queries.

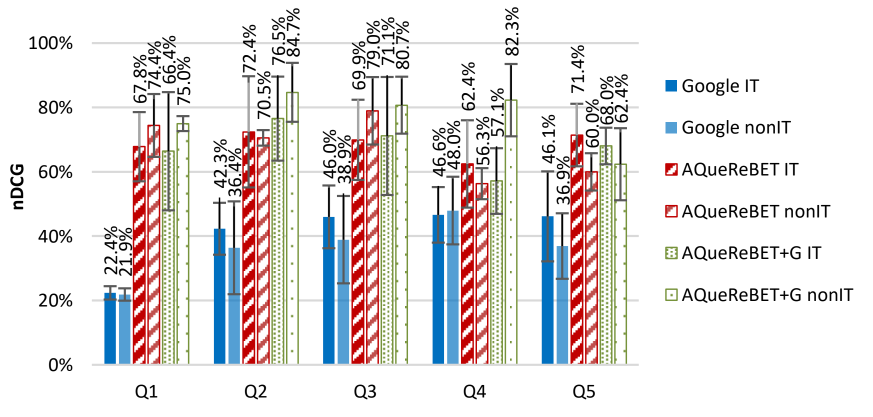


Figure 8. IT and nonIT comparison of nDCG metric for results provided by Google, AQueReBET and AQueReBET + G

We are aware of the desirability of comparison with related works. Since there are only a few works in this area and there is no common adopted methodology or dataset, it is not possible to compare directly different approaches and their results. We hope, that we have fulfilled the need for comparison at least partially by adopting Google as a baseline (which is and will be available to any other researcher in the future).

6 CONCLUSIONS

In this paper we described a new method called AQueReBET – automatic query refinement based on eye-tracking. This method provides web seekers during their web search (using their implicit feedback) with more relevant documents. We chose the way of extracting possible new words from the snippets in the page of SERP, where each word-level importance is calculated based on the term frequency, term uniqueness (tf-idf) and total fixation duration within the snippets. We evaluated the proposed approach in a study with 22 participants. The results support our hypothesis that the information obtained from the analysis of the seeker's gaze can improve the relevance of the pages/documents provided to the seeker. On average, our solution improved nDCG@5 of web searches from 38 % to 74 % alternatively 78 %, representing a rise by 36 alternatively 40 percentage points as well as a 1.9 alternatively 2.1 times increase. We managed to reduce the query reformulation rate to approximately 23.6 % on average. The most notable improvement was obtained for the most ambiguous query Q1. Considering several related works on gaze-based feedback, improvement has also been reported by other authors [5, 8, 3, 31].

When making a quantitative comparison with [5], they achieved (for $k = 5$ and similar baseline) a 1.37-fold improvement. Their maximum value of nDCG@k is 40.8 %, whereas ours is 79.5 %. This is, even respecting the 13.7 % standard deviation, considerably better. They used also MAP metrics (mean average precision). We have an improvement from MAP@5 = 44 % to 93 %, representing a rise of 49 percentage points as well as 2.12 times increase. They improved their absolute MAP from 46.6 % to 55.9 % representing a rise of 9.3 percentage points as well as 1.20 times increase. Since our and their baseline results are very close, we consider our results considerably better. However, it should be noted that the datasets were not the same, and nor were the experiment setting and methodology.

Buscher et al. [6] performed a very similar experiment. We consider their results only regarding a read length of 150 characters because our snippets are cca 150 characters long. They achieved DCG@10 = 9.15 which corresponds to nDCG@10 = 29.2 %. Our result is nDCG@9 = 70 % which is 2.4 times better. They achieved MAP@10 = 73.6 %, our result of 93 % makes us 1.26 times better.

Umemoto et al. [31] also used similar metrics, in particular nDCG and MAP to evaluate their results. They achieved nDCG = 81.6 %, our best value of nDCG@2 = 79.5 %. The difference is probably not statistically significant. They achieved MAP = 65.2 % and we achieved 94.5 %, what is 1.45 times better. However these results have been achieved under slightly different conditions so they can serve for a rough comparison only. The main difference in evaluation is that they considered up to 15 terms whereas we considered 4 words and their rating scale had 3 degrees whereas ours 11, and their binary mapping of user evaluations uses a different threshold than does ours.

Eickhoff et al. [8] used MRR metric attaining values 0.80 and 0.86, compared with 0.79 MRR value for Buscher et al. [5]. In our method, MRR metric for relevant results (seeker's evaluation value 7 or higher) is 0.87 for AQueReBET and 0.91 for AQueReBET + G.

In the experiments, we also studied the effect of groupization. The groupization improved the nDCG@5 on average by 4 percentage points, although the improvement was not significant. In general, however, we see for this approach a potential for improvement. E.g. bearing in mind the quick response of our system, our calculations worked only with the first 10 Google results. If we would take more of them, the nDCG would be even higher. One of the challenges is that it would sometimes benefit from an ability to choose proper query words based on their meaning (semantics).

The other important result of our research, besides the above mentioned method, is also the automatic objective evaluator, which allows us to order the results in SERP and get such high results in nDCG, MAP and MRR. We determined its accuracy and obtained 75 %. We consider it high, keeping in mind that seekers' subjective evaluations differ a little bit and thus 100 % accuracy is not achievable.

Results of our comparison of IT vs nonIT groups suggest that groupization is helpful especially for seekers that have lower web searching skills. However, this requires further research with more participants and more queries.

It should be observed that our approach works when at least one of the results in SERP corresponds to a user's intention. In the opposite case, i.e., when all results do correspond, there is no need to apply our method. We identify these to be limitations of our work.

Future work and possible improvements lie in creating an ontology with which we could extract appropriate words in a better way. Some use Wikipedia as a source of semantics [9]. Using semantic links between words we would be able to remove unrelated words from the table, resulting in more accurate suggestions. Enhancing semantic approach by sentiment could provide possibly even better results [1].

Another direction of future work may be to add a fourth assumption (and incorporate it into our Equation (3)): The higher the importance of a word, the better position it has within the snippet – e.g. the closer to a queried word, the higher the importance.

A different possible line of research of utilization of eye tracking feedback could be inspired by recent progress in the related research on utilization of cursor movement data [18], where frequent subsequences called motifs are automatically discovered to be used to improve result relevance estimation and re-ranking.

Acknowledgments

This work was partially supported by the Scientific Grant Agency of Slovakia, grant No. VG 1/0667/18 and it is a partial result of the development project No. 002STU-2-1/2018 funded by the Ministry of Education, Science, Research and Sport of the Slovak Republic. It was also supported by the Research and Development Support Agency, grant No. APVV-15-0508 and grant No. APVV-17-0267.

REFERENCES

[1] Abdi, A.—Shamsuddin, S. M.—Aliguliyev, R. M.: QMOS: Query-Based Multi-Documents Opinion-Oriented Summarization. Information Processing and Management, Vol. 54, 2018, No. 2, pp. 318–338, doi: 10.1016/j.ipm.2017.12.002.

[2] Anderson, L. W.—Krathwohl, D. R.—Bloom, B. S.: Taxonomy for Learning, Teaching, and Assessing: A Revision of Bloom's Taxonomy of Educational Objectives. Longman, New York, 2001.

[3] Biedert, R.—Dengel, A.—Käding, C.: Universal Eye-Tracking Based Text Cursor Warping. Proceedings of the Symposium on Eye Tracking Research and Application (ETRA '12), ACM, 2012, pp. 361–364, doi: 10.1145/2168556.2168637.

[4] Bing, L.—Lam, W.—Wong, T.-L.—Jameel, S.: Web Query Reformulation via Joint Modeling of Latent Topic Dependency and Term Context. ACM Transactions on Information Systems, Vol. 33, 2015, No. 2, Art. No. 6, 38 pp., doi: 10.1145/2699666.

[5] Buscher, G.—Dengel, A.—van Elst, L.: Query Expansion Using Gaze-Based Feedback on the Subdocument Level. Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '08), ACM, 2008, pp. 387–394, Art. No. 6, 38 pp., doi: 10.1145/2699666.

[6] Buscher, G.—Dengel, A.—Biedert, R.—van Elst, L.: Attentive Documents: Eye Tracking as Implicit Feedback for Information Retrieval and Beyond. ACM Transactions on Interactive Intelligent Systems, Vol. 1, 2012, No. 2, Art. No. 9, 30 pp., doi: 10.1145/2070719.2070722.

[7] Carpineto, C.—Romano, G.: A Survey of Automatic Query Expansion in Information Retrieval. ACM Computing Surveys, Vol. 44, 2012, No. 1, Art. No. 1, 50 pp., doi: 10.1145/2071389.2071390.

[8] Eickhoff, C.—Dungs, S.—Tran. V.: An Eye-Tracking Study of Query Reformulation. Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '15), ACM, 2015, pp. 13–22, doi: 10.1145/2766462.2767703.

[9] Goslin, K.—Hofmann, M.: A Wikipedia Powered State-Based Approach to Automatic Search Query Enhancement. Information Processing and Management: An International Journal, Vol. 54, 2018, No. 4, pp. 726–739, doi: 10.1016/j.ipm.2017.10.001.

[10] Granka, A. L.—Joachims, T.—Gay, G.: Eye-Tracking Analysis of User Behavior in WWW Search. Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '04), ACM, 2004, pp. 478–479, doi: 10.1145/1008992.1009079.

[11] Huang, J.—White, R.—Buscher, G.: User See, User Point: Gaze and Cursor Alignment in Web Search. Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '12), ACM, 2012, pp. 1341–1350, doi: 10.1145/2207676.2208591.

[12] Järvelin, K.—Kekäläinen, J.: Cumulated Gain-Based Evaluation of IR Techniques. ACM Transactions on Information Systems, Vol. 20, 2002, No. 4, pp. 422–446, doi: 10.1145/582415.582418.

[13] Joachims, T.—Granka, T.—Pan, B.—Hembrooke, H.—Radlinski, F.—Gay, G.: Evaluating the Accuracy of Implicit Feedback from Clicks and Query Reformulations in Web Search. ACM Transactions on Information Systems, Vol. 25, 2007, No. 2, Art. No. 7, doi: 10.1145/1229179.1229181.

[14] Spärck Jones, K.: A Statistical Interpretation of Term Specificity and Its Application in Retrieval. Journal of Documentation, Vol. 28, 1972, No. 1, pp. 11–21, doi: 10.1108/eb026526.

[15] Kajan, R.—Herout, A.—Bednarik, R.—Povolný, F.: PeepList: Adapting Ex-Post Interaction with Pervasive Display Content Using Eye Tracking. Pervasive and Mobile Computing, Vol. 30, 2016, Issue C, pp. 71–83, doi: 10.1016/j.pmcj.2015.12.004.

[16] Kompan, M.: Group and Single-User Influence Modeling for Personalized Recommendation. Information Sciences and Technologies Bulletin of the ACM Slovakia, Vol. 6, 2014, No. 2, pp. 11–20.

[17] Kovárová, A.: Special Interaction Approaches and Their Impact on Usability. Information Sciences and Technologies Bulletin of the ACM Slovakia, Vol. 3, 2011, No. 3, pp. 14–25.

[18] Lagun, D.—Ageev, M.—Guo, Q.—Agichtein, E.: Discovering Common Motifs in Cursor Movement Data for Improving Web Search. Proceedings of the 7th ACM International Conference on Web Search and Data Mining (WSDM '14), ACM, 2014, pp. 183–192, doi: 10.1145/2556195.2556265.

[19] Li, Q.—Tian, M.—Liu, J.—Sun, J.: An Implicit Relevance Feedback Method for CBIR with Real-Time Eye Tracking. Multimedia Tools and Applications, Vol. 75, 2016, No. 5, pp. 2595–2611, doi: 10.1007/s11042-015-2873-1.

[20] Loboda, T. D.—Brusilovsky, P.—Brunstein, J.: Inferring Word Relevance from Eye-Movements of Readers. Proceedings of the 16th International Conference on Intelligent User Interfaces (IUI '11), ACM, 2011, pp. 175–184, doi: 10.1145/1943403.1943431.

[21] Low, T.—Bubalo, N.—Gossen, T.—Kotzyba, M.—Brechmann, A.—Huckauf, A.—Nürnberger, A.: Towards Identifying User Intentions in Exploratory Search Using Gaze and Pupil Tracking. Proceedings of the 2017 Conference on Human Information Interaction and Retrieval (CHIIR '17), ACM, 2017, pp. 273–276, doi: 10.1145/3020165.3022131.

[22] Maglio, P. P.—Barrett, R.—Campbell, C. S.—Selker, T.: SUITOR: An Attentive Information System. Proceedings of the 5th International Conference on Intelligent User Interfaces (IUI '00), ACM, 2000, pp. 169–176, doi: 10.1145/325737.325821.

[23] Manning, C. D.—Raghavan, P.—Schütze, H.: Introduction to Information Retrieval. Cambridge University Press, 2008.

[24] Móro, R.—Daráž, J.—Bieliková, M.: Visualization of Gaze Tracking Data for UX Testing on the Web. Late-Breaking Results – Doctoral Consortium and Workshop Proceedings of the 25th ACM Hypertext and Social Media Conference (Hypertext 2014 – Extended Proceedings), CEUR Workshop Proceedings, Vol. 1210, 2014.

[25] Murata, M.—Toda, H.—Matsuura, Y.—Kataoka, R.: Query-Page Intention Matching Using Clicked Titles and Snippets to Boost Search Rankings. Proceedings of the 9th ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL '09), ACM, 2009, pp. 105–114, doi: 10.1145/1555400.1555419.

[26] Papoutsaki, A.—Laskey, J.—Huang, J.: SearchGazer: Webcam Eye Tracking for Remote Studies of Web Search. Proceedings of the 2017 Conference on Human Information Interaction and Retrieval (CHIIR '17), ACM, 2017, pp. 17–26, doi: 10.1145/3020165.3020170.

[27] Spink, A.—Jansen, B. J.—Cenk Ozmultu, H.: Use of Query Reformulation and Relevance Feedback by Excite Users. Internet Research, Vol. 10, 2000, No. 4, pp. 317–328, doi: 10.1108/10662240010342621.

[28] Teevan, J.—Ringel Morris, M.—Bush, S.: Discovering and Using Groups to Improve Personalized Search. Proceedings of the Second ACM International Conference on Web Search and Data Mining (WSDM '09), ACM, 2009, pp. 15–24, doi: 10.1145/1498759.1498786.

[29] Tran, V. T.—Fuhr, N.: Using Eye-Tracking with Dynamic Areas of Interest for Analyzing Interactive Information Retrieval. Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '12), ACM, 2012, pp. 1165–1166, doi: 10.1145/2348283.2348521.

[30] White, R. W.—Chu, W.—Hassan, A.—He, X.—Song, Y.—Wang, H.: Enhancing Personalized Search by Mining and Modeling Task Behavior. Proceedings of the 22nd International Conference on World Wide Web (WWW '13), ACM, 2013, pp. 1411–1420, doi: 10.1145/2488388.2488511.

[31] Umemoto, K.—Yamamoto, T.—Nakamura, S.—Tanaka, K.: Search Intent Estimation from User's Eye Movements for Supporting Information Seeking. Proceedings of the International Working Conference on Advanced Visual Interfaces (AVI '12), ACM, 2012, pp. 349–356, doi: 10.1145/2254556.2254624.

[32] Umemoto, K.—Yamamoto, T.—Nakamura, S.—Tanaka, K.: Predicting Query Reformulation Type from User Behavior. Proceedings of the 28th Annual ACM Symposium on Applied Computing (SAC '13), ACM, 2013, pp. 894–901, doi: 10.1145/2480362.2480534.

[33] Wildemuth, B. M.—Kelly, D.—Boettcher, E.—Moore, E.—Dimitrova, G.: Examining the Impact of Domain and Cognitive Complexity on Query Formulation and Reformulation. Information Processing and Management, Vol. 54, 2018, No. 3, pp. 433–450, doi: 10.1016/j.ipm.2018.01.009.

[34] Xu, S.—Jiang, H.—Lau, F. C. M.: Personalized Online Document, Image and Video Recommendation via Commodity Eye-Tracking. Proceedings of the 2008 ACM Conference on Recommender Systems (RecSys '08), ACM, 2008, pp. 83–90, doi: 10.1145/1454008.1454023.

[35] Zhu, G.—Mishne, G.: ClickRank: Learning Session-Context Models to Enrich Web Search Ranking. ACM Transactions on the Web, Vol. 6, 2012, No. 1, Art. No. 1, 22 pp., doi: 10.1145/2109205.2109206.

Alena Martonova received her computer science M.Sc. degree from the Comenius University in Bratislava in 2004, and Ph.D. degree from the Slovak University of Technology in Bratislava in 2011. Her research interests involve human computer visualization and visualization. She is currently Assistant Professor at the Faculty of Informatics and Information Technologies, Slovak University of Technology in Bratislava.

Jozef Marcin received his software engineering B.Eng. and M.Eng. degrees from the Faculty of Informatics and Information Technologies, Slovak University of Technology in Bratislava in 2013 and 2015, respectively. His research interests combine web applications, web search and eye tracking. He currently works as a software engineer in industry, he has developed infrastructure for IoT and Industry 4.0.

Pavol Navrat received his M.Sc. and Ph.D. degrees from the Slovak University of Technology in Bratislava in 1975 and 1983, respectively. He (co-)authored several books and numerous scientific papers. His research involves intelligent information systems, software engineering and artificial intelligence. He is currently Full Professor at the Faculty of Informatics and Information Technologies, Slovak University of Technology in Bratislava.

Jozef Tvarozek received his computer science M.Sc. degree from the Comenius University in Bratislava in 2007, and Ph.D. degree from the Slovak University of Technology in Bratislava in 2011. In research, he is interested in eye-tracking and online learning of programming via active problem solving. He is currently Assistant Professor at the Faculty of Informatics and Information Technologies, Slovak University of Technology in Bratislava.

Gabriela Grmanova received her computer science M.Sc. degree from the Comenius University in Bratislava in 1998, and Ph.D. degree from the Slovak University of Technology in Bratislava in 2005. Her research interests involve data mining, pattern recognition and mathematical modelling. She is currently Assistant Professor at the Faculty of Informatics and Information Technologies, Slovak University of Technology in Bratislava.

Computing and Informatics, Vol. 38, 2019, 1375–1402, doi: 10.31577/cai_2019_5_1375

FRAMEWORK FOR KNOWLEDGE DISCOVERY IN EDUCATIONAL VIDEO REPOSITORIES

Jorão Gomes Jr., Laura Lima Dias, Eduardo Rocha Soares

Postgraduate Program in Computer Science
Federal University of Juiz de Fora
36036-900, Minas Gerais, Brazil
e-mail: {joraojunior, laura.lima, eduardosoares}@ice.ufjf.br

Eduardo Barrere, Jairo Francisco de Souza

LApIC Research Group, Department of Computer Science
Institute of Exact Sciences, Federal University of Juiz de Fora
36036-900, Minas Gerais, Brazil
e-mail: {eduardo.barrere, jairo.souza}@ice.ufjf.br

Abstract. The ease of creating digital content coupled with technological advancements allows institutions and organizations to further embrace distance learning. Teaching materials also receive attention, because it is difficult for the student to obtain adequate didactic material, being necessary a high effort and knowledge about the material and the repository. This work presents a framework that enables the automatic metadata generation for materials available in educational video repositories. Each module of the framework works autonomously and can be used in isolation, complemented by another technique or replaced by a more appropriate approach to the field of use, such as repositories with other types of media or other content.

Keywords: Semantic annotation, knowledge discovery, video repositories

Mathematics Subject Classification 2010: 68-P20

1 INTRODUCTION

The ease of creating digital content coupled with technological advances enables institutions and organizations to increasingly adopt distance learning [16, 41]. Current efforts at distance education are geared towards a more individualized and personalized education. Researchers are interested in observing and modeling the profile of students, making it possible to adapt the learning according to the personality and the needs of students [33]. This factor allows the effective use of distance education systems and the permanence of students in the offered courses [6].

Another factor that also contributes to this effective use of e-learning systems is the correct administration of didactic materials, in order to make them available according to the learning needs of each student [42]. Teaching materials have also received attention from the researchers, since it is difficult for a student to obtain an adequate learning material by himself. A high effort and previous knowledge about the material and the repository are necessary to be succeeded in the search task. This difficulty is further compounded by the growth in the number of learning materials in the repositories [26], which can cause many irrelevant materials to be returned in the search. The fact that, even with advances in technology, students still cannot obtain what they want by doing searches in web repositories indicates the relevance of the studies focused on the understanding of these repositories in order to improve the search and the administration of the didactic materials, especially when we talk about videos the natural language of which is often vague and uncertain [15].

In addition, it must be kept in mind that most of the time spent on learning online courses is dominated by student interaction, unlike in-person courses where the instructor dominates most of the time. Another point to note is that the classroom is a network in which students from different geographic locations interact socially, sharing information and resources, and even performing joint projects [20]. Therefore, for a correct administration of learning materials, it is also necessary to consider an adequate creation of these shared information spaces. The work of [34], for example, presents a good alternative for the creation of globally shared information spaces, named the Linked Data initiative. This initiative is an interesting option for the discovery and understanding of Open Educational Resources (OER) data. Linked Data consists of a set of practices for publishing, sharing and interconnecting data in Resource Description Framework (RDF) format. Educational repository administrators are realizing the potential of using Linked Data for describing, discovering, linking and publishing educational data on the Web [34]. The Linked Data is based on the use of Uniform Resource Identifier (URI) references to identify digital documents, as well as real content and even abstract concepts [22]. Thereby, obtaining materials that use this format allows a greater flexibility and connection within the repositories of learning objects, as is the case of the works [29, 9].

This work proposes the creation of a framework that enables the generation of metadata for the materials available in educational repositories of videos and texts, in order to facilitate the creation of these shared information spaces. In addition, this research intends to present a way for administrators of repositories and course creators to better know their repository as a whole, through a panoramic view of the areas of concentration. In this work, we consider scenarios for the use of this framework on repositories with materials that have little or no previously associated metadata. A case study of the use of the framework on a real repository of educational videos will also be presented. Besides, the present work shows how didactic materials are related within the repository at the end of the process. These relationships give a big picture of how the areas of knowledge present in video lectures are related.

2 RELATED WORK

Many works in the literature relate to ways to better explore and gather information about the media contained in repositories. Metadata are important for integrating learning objects from different repositories, and recent integration strategies are heavily dependent on metadata previously associated with learning objects. For example, in [7, 40], the authors present a recommendation system for the Moodle platform that indexes learning objects from different repositories and searches in their stored metadata. A re-design of the Moodledata module functionalities is presented in [10]. The authors aim to share learning objects between e-learning content platforms, e.g., Moodle and G-Lorep, in a linkable object format. Their proposal allows a semantic description of the learning objects. However, we argue in this paper that metadata of learning objects in open repositories, especially video lectures, are usually poorly descriptive. In Section 3.1.1 we present an analysis of metadata quality in an open Brazilian video lecture repository. Automatic metadata extraction techniques are important to enrich learning object information. In [38], natural language processing techniques are used to improve browsing and searching within the BBC's radio program repositories. Some papers in the literature also rely on technologies such as Automatic Speech Recognition (ASR), that is the process that allows computers to receive a speech signal as input and convert it into natural language text. This technique is used to automatically extract information from the audio track of multimedia content on the web. The extracted information, generally, has two main uses: in the composition of multimedia applications (e.g., closed caption, personal assistants, other ways of accessibility) or in understanding multimedia content in order to improve the search for it on the web [44, 3]. In the second case, ASR has great importance on extracting content from an audio signal that can be useful for media representation. Researches such as that of [44], for example, make use of ASR to extract the spoken content of videos in order to extract keywords through semantic annotation. Then, the extracted keywords are used to recommend those videos through similarity calculations. The present work, however, uses the result of ASR as input to a series of natural language processing techniques to associate semantic resources with educational videos in order to help to identify videos with similar content and calculate similarity to discover and understand the repository.

Concepts addressed in the materials of a repository can be discovered and retrieved by several processes. Other works have also explored the task of automatic indexing or automatic annotation, as in [21, 44, 45]. In [21], the authors demonstrate the effect of extracting and combining visual and audio information into the search process using part of the TREC 2001[1] Video Recovery Trail for evaluation. Among the analyzed information are: speech recognition, face detection, text extraction via OCR and the use of image similarity matching. OCR techniques are also used in [25] to summarize fixed-camera video lectures by detecting handwritten content. In [45] the authors present a visual navigation system for exploring bio-medical OER videos, while the present work explores linked data for the discovery of the main topics and relations among videos. Word Embedding models were used in [11] to detect segment boundaries in video lectures. The authors argue that classical scene detection algorithms are useless for segmenting video lectures because this kind of video is usually recorded in one shot.

The calculation of similarity between media is used by some systems to support the content recommendation process. Iris AI[2], for example, makes the recommendation of scientific articles from an initial user's indication. Like this work, Iris AI calculates the similarity between documents through the relations in knowledge bases. However, this is done with the focus on the recommendation of the articles, while our work is also concerned with the existing knowledge in the repositories where the video lectures are stored. In [30], the authors closely approximate our research by employing ontologies and automatic metadata annotation, retrieving information to use recommendations and also by aggregating related content. However, the results are still experimental and limited to the relations defined in the SCORM standard.

The work of [12] reports that one of the main challenges in the implementation of technological services in repositories is the visualization of information and that current research in this area is directed towards the improvements in the retrieval of scientific and academic information. Finally, the works [13, 35] report that new orientation strategies for service innovation and the functionality of technological means in institutional repositories are needed, as well as the effort to solve problems such as obsolescence and guaranteeing the satisfaction of academic communities based on the usefulness of the repositories and the experience and usability of students and users. So [13] builds a systematic review about the user experience in institutional repositories and discusses possible solutions to collaborate with those needs.

Considering the recent and important concerns with the use and knowledge of educational repositories, our work aims to make it possible to learn about educational video repositories, providing automatic metadata to support their usability. We present a framework used to extracted metadata from video lectures based on speech information. We compare different techniques used for automatic semantic

[1] `https://trec.nist.gov/`

[2] `http://iris.ai`

annotation. Unlike other works, we present how the extracted semantic metadata can be used in two tasks: similarity calculation and clustering. In similarity calculation, we propose an algorithm to extract and compare extrinsic metadata information provided by the DBpedia ontology. In the clustering task, we show how the label propagation algorithm can be used on the knowledge graph to identify items in similar domains. We make all data available for further research.

3 CONCEPTS AND TECHNIQUES

Currently, several techniques can be used for video indexing, such as color histogram sorting, shape recognition, action recognition, face recognition, text extraction through Optical Character Recognition (OCR), among others. Regardless of how data is collected from videos, these videos can be associated with pre-existing concepts from a knowledge base. These concepts can be seen as the main topics. This process is called semantic annotation. The semantically annotated videos are related to entities, which, in turn, are part of a network of relationships with meanings, such as an ontology or a thesaurus. Thus, media annotation facilitates the search and recommendation processes in repositories and many researchers have been working on improvements in it for various media [5, 36].

The following sections aim to clarify the concepts and techniques used in the proposed framework. They are organized to present the process of indexing and retrieving information, starting from video files until the relationship of the videos with close contents.

3.1 Process Architecture

It is a difficult task for students to find satisfactory didactic materials. This activity demands effort and knowledge about the material and the repository. The difficulty is worsened by the growth in the number of learning materials [26]. Therefore, educational video repositories must deal with these difficulties, especially when they are constantly fed with new materials.

Since the proposal of this work is to facilitate the understanding of these materials and repositories, we use as the setting for our experiments a real repository of educational videos produced by Brazilian universities. The application scenario is contextualized within the Video Advanced Search Group (GT-BAVi) of the Brazilian National Research and Educational Network[3] (RNP). The objective of the GT-BAVi is to develop a prototype to facilitate the semantic enrichment of the video repository and to facilitate the search. For this, a framework was implemented in order to accomplish this task automatically.

The developed solution can be divided into three main steps: Content Processing, Context Association, and Knowledge Graph. For the Content Processing step, an ASR system was used, since the main focus of the framework is on processing

[3] `https://www.rnp.br`

videos. We can, then, extract the video lecture content in a text format through its audio track. For the Context Association step, we used an Automatic Semantic Annotation method to attach concepts from a knowledge base to the videos. Finally, in the Knowledge Graph step, we obtain the relationships between the videos according to their similarity related to the extracted concepts.

Figure 1 represents the solution steps. Each step is implemented by a corresponding process.

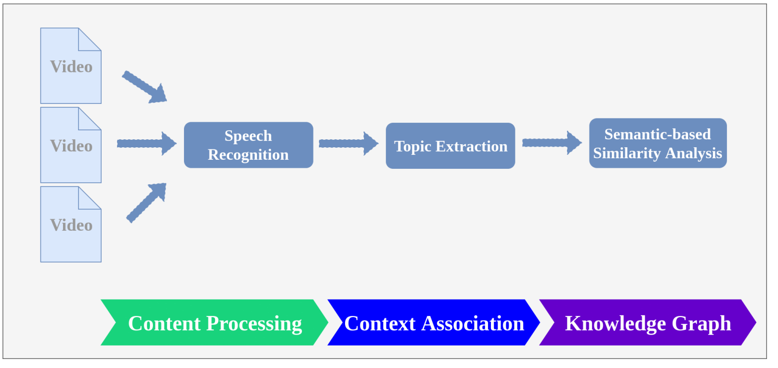


Figure 1. Processing Framework

3.1.1 Analysis of the RNP Repository

Among all the RNP repositories, the video repository VideoAula@RNP[4] is the focus of our research, since it is currently used by Brazilian academic institutions to store video lectures. An analysis of the pre-existing video metadata in the RNP repository was performed. This analysis is relevant to an initial validation of the prototype. These metadata are tags manually created by video editors to ensure that the video is found through keyword searches. The information collected for this analysis was the number of videos within the repository, the total amount of metadata used in the repository, how many of them were different, how many tags on average each video received, the number of videos with useless tags, as tags out of context or very generic. Examples of useless tags are “video”, “video lectures”, “tag”, “test”, “teacher description”. The information collected is in Table 1.

The RNP repository had 858 video lectures. Each video had an average of 2 to 3 metadata, which totaled 2 225 metadata in the repository. However, only about a third of them were unique, indicating that many of the metadata were repeated. Thus, many videos would be returned after searching for some keywords. Furthermore, 604 videos had useless metadata, i.e., they did not add a specific identification to the videos to which they were attached.

[4] `http://www.videoaula.rnp.br`

Information	**Collected Values**
Number of videos	858
Total of tags	2 225
Total of distinct tags	849
Average tags per video	2.59 ± 1.34
Number of videos with useless tags	604
Number of videos that did not have tags	2

Table 1. RNP scenario using only manually associated metadata

After collecting more specific information about the metadata, the following data were also found: 540 videos had only 2 metadata, being "video" and "video lectures", making it impossible to identify any of these videos by their content. One of the videos had only the "test" metadata. Still, 16 videos had only the "Teacher description" metadata. These data show that it is impossible to find a specific subject in any of these videos just by searching for terms. The data collected also indicates that the metadata attached to the video often limit the potential of a search in the repository. This situation occurs mainly due to informalism and little dedication during the metadata creation stage when uploading videos, generating inappropriate metadata for a future video search.

Since many videos in the repository have too few metadata, and their titles are generally vague (e.g. "Exercise_5"), this repository is a good choice to show the potential of our proposal based on the ASR and semantic annotation. We want to show that our framework is able to automatically extract meaningful information that can be used to improve the search and recommendation of these videos from which we previously had no information.

3.2 Description of the Framework

We have defined the framework as a process composed of three steps that work in isolation and these steps will be validated individually. Since the audio is the main source of information in video lectures, the Subsection 3.2.1 presents an ASR system trained for this work. The Subsection 3.2.2 presents the Context Association supported by Natural Language Processing techniques that allow automatic semantic annotation. In this section we will present two options for this task and discuss the results for both. Finally, the Subsection 3.2.3 presents the Knowledge Graph supported by the similarity calculation between videos by walking in the category graph of a knowledge base. In addition, the parameters used in this walk are discussed in order to demonstrate the best options considering the accuracy and computational cost.

The differential point of the process presented in this work is the possibility to develop the process steps as different services that can be instantiated as needed, such as the ones regarding the need of specific processing steps and the type of material utilized.

3.2.1 Content Processing

The automatic semantic annotation process discussed in this paper focused on the video lectures scenario. In this context, even more than in others, most of the information is present in the teacher's speech. For this reason, the automatic semantic annotation process depends primarily on ASR. According to [14], ASR systems generally are built on three main models: **acoustic**, **lexical** and **language model**.

The acoustic model is responsible to allow the ASR system to determine which sequences of speech unit (generally, phonemes) have more similarity with the vectors of acoustic characteristics that were extracted from the audio signal. Thus, the acoustic modeling is done by training an algorithm to predict the probabilities of phonemes to be related to an audio segment. Some of the most popular algorithms to do acoustic modeling are the Hidden Markov Models (HMMs) [18] and Deep Neural Networks (DNNs) [24]. For training an acoustic model, it is necessary to provide a corpus of speech containing well segmented audio files with speeches from the specific target for which the ASR is being designed. Furthermore, the training also requires their respective ground-truth transcriptions.

A lexical model is basically a dictionary that maps words of a vocabulary to a sequence of phonemes. This dictionary is used by the ASR system so that it can convert the sequences of phonemes recognized through the acoustic model into words. To create this model, there are phonetic converters that take a sequence of characters and return a sequence of correspondent phonemes. For example, in [39], the authors have used Long Short-Term Memory (LSTM) recurrent neural networks to do the automatic grapheme-to-phoneme conversion.

The language model is not essential for the operation of an ASR system. However, its use significantly improves the accuracy of those systems because the acoustic model is not enough to obtain a satisfactory transcription. The acoustic model only infers sequences of phonemes that are then converted into a sequence of words, through the lexical model, without any grammatical restrictions. The language model acts exactly on this issue by calculating conditional probabilities of words from the vocabulary to be recognized after others. With this, it is possible to restrict the possibilities of recognized sequences of words. Thus, ungrammatical sentences have low probability to be formed, and that reduces the search space, decreases the time for recognition, and improves acoustic ambiguities resolution [43]. Therefore, it is a consensus that in systems which deal with wide vocabularies, like the ASR system for continuous speech that is used in this work, the language model is extremely important. The training of the language model is done through a text corpus where the word frequencies and conditional probabilities of the word sequences are extracted.

To create a robust ASR system it is important that the system is trained with a large data volume that covers the main characteristics and variations (e.g. noise in the audio, accent, intonation, gender, age) present on the speeches of the system's target public. The biggest challenge in designing ASR systems is to obtain a speech and text corpora that are proper for training. The creation of these training bases is a costly process, as there is not enough free and open properly catalogued audio samples available, and the process of creating such samples is expensive in terms of time, space and money. When we talk specifically about training ASR systems for Brazilian Portuguese, that difficulty is even bigger, which makes those systems perform poorly when dealing with different accents and with different types of noise and distortion in the speech signal [32]. An alternative to obtaining a better accuracy in ASR with few data is to train it to be a specialized system by using databases that contemplate only the target scenario of the final application.

That is why in this work we trained our own specialized ASR system for Brazilian Portuguese video lectures. To train the acoustic model, we use a speech corpus extracted from subtitled video lectures in Brazilian Portuguese that are made available for free by Coursera[5], with a total of 55 hours of audio. We also added to our dataset a total of more than 2 hours of audio from corpora made available for free by VoxForge project[6] and by the Signal Processing Laboratory of UFPA (LAPS-UFBA) from Brazil[7]. The acoustic model is based on Deep Neural Network (DNN), which has 440 neurons in the input layer and 6 hidden layers of 2048 neurons each. The output layer has around 4000 neurons. For language model training, we use a union of text corpora from multiple sources such as subtitles from Coursera video lectures, open and free text corpora like CETEN, OGI and LapsFolha that are also made available by the LAPS plus Wikipedia articles. In total, the text corpus used to train our language model has about 13 million sentences.

To obtain the final ASR model that we use in the framework proposed in this work, we performed experiments aiming to explore different training configurations and pre-processing steps, separately or combining them, in order to verify which of them are responsible for obtaining a complete ASR model that has better recognition accuracy in our test data. For the acoustic model, we evaluate the impact of changing the training algorithm and audio sample rate. We have also evaluated the impact of segmenting the audio into smaller chunks and aligning them with their respective transcriptions. For the language model, we evaluated the effects of the variation of the model order[8], text pre-processing and probability smoothing methods.

To evaluate the trained models, we build manually an evaluation dataset composed of 2 hours of audio extracted from different parts of video lectures that are not in our training data. To build this evaluation dataset, we transcribed manually

---

[5] `https://pt.coursera.org/`

[6] http://www.voxforge.org/home

[7] `https://laps.ufpa.br`

[8] The order is related to the dependence of each word given the $n-1$ words which precede it. This means that for a model of order 3 (trigram), the probability of occurrence of a word is related to the two that precede it. For the order 4 (4-gram) model, the probability of occurrence of a word is related to the three that precede it, and so forth.

each audio in it, which resulted in about 581 spoken sentences with a ground-truth transcription. The accuracy metric used to evaluate the models was the Word Error Rate (WER). This metric represents the number of modifications (insertion, replacement or removal of words) that are necessary for the recognized sentences to transform them into the correct ones. That is, the lower its value, the better the accuracy of the recognizer [32].

After the experimentation, we get the best WER of 45.5 % with the following training settings:

- The training of the **best acoustic model** was done using the WAV codec, audio sample rate of 8000 Hz, MONO channel, and DNNs as the training algorithm. In this model, we have also performed the segmentation and alignment of audios with their respective transcriptions.
- The best language model was obtained with the interpolation of two other models, of 4-gram and 3-gram, with a normalized training corpus. The following tasks were applied in the normalization step: setting all words to lowercase; transforming all dates, times, percentages, Roman numerals, cardinal and ordinal numbers, acronyms, abbreviations and monetary values to their full forms (e.g., if it was in English, "5°" and "100 %" would become "fifth" and "one hundred percent", respectively). Furthermore, the applied probability smoothing was the Kneser-Ney method [31].

The WER of 45.5 % in speech recognition that we obtained in this work are close to those obtained in commercial systems such as Google[9], Microsoft[10] and IBM[11]. For our video lectures evaluation dataset, Google obtained 35.9 %, IBM 73.7 % and Microsoft's model achieved a result of 44.7 %.

Since ASR is used as the basis for the following processes of our framework, it is important that the recognition error rate be the smallest possible. The presented results in this subsection showed that our specialized ASR model is capable of obtaining a good performance in the context of this work. However, it is still not an error-free process. Therefore, in the following subsections we analyze and discuss ways of performing the following processes on the noisy video lecture transcriptions from ASR.

3.2.2 Context Association

There are several approaches aiming at video search improvement through semantic annotation. The approaches to assign these annotations can be divided between those that make use of external data related to the video and those that use only the information contained in the media. In the first group, for example, the use of

[9] https://cloud.google.com/speech/
[10] https://docs.microsoft.com/pt-br/azure/cognitive-services/Speech/API-Reference-REST/BingVoiceRecognition
[11] https://www.ibm.com/watson/developercloud/speech-to-text.html

texts around images is used by [17] to verify correspondences between images and texts and thus to find the interrelated sets of terms and topics, instead of simply annotating texts. For works that only use information contained in the media, we can cite the work of [1], which presents some approaches with event-based content (using the visual content of the videos). Among the approaches presented, there are mechanisms such as limit detection of takes, keyframes extraction for representation of important parts of the video, structural analysis and scene segmentation combined with OCR techniques for extraction of textual resources and creation of tags. Although these approaches have good classification results, they are limited to specific types of videos, usually those with a well-defined content and temporal structure.

Although there are several approaches for video annotation, there are not many studies that analyze the quality of information used to semantically annotate educational videos. Textual quality information is commonly present in video lectures repositories. For example, almost all Videoaula@RNP videos were recorded as an expository lesson, containing slide projection throughout the video. However, most of the information content of the video is in the teacher's speech. Therefore, search engines that only use video metadata (title or abstract) cannot help the user when he/she wants to find a video using terms that appeared during an exercise or an example that the teacher approached. In this view, we analyzed the impact of the semantic annotation process on several data sources extracted from the Videoaula@RNP and how the quality of the new tags created can influence the knowledge about the video lectures; for use of the search engines, for example, and general knowledge of the repository.

The automatic semantic annotation experiment was performed using two approaches: an Entity Linking Approach and a Topic Extraction Approach. The Entity Recognition Approach has a natural language text as input and produces a set of $(term, entity)$ pairs that represent the concept (entity) associated with the term present in the text. For this process, natural language processing techniques are typically used to tokenize the text and identify the correct terms. We used DBpedia Spotlight [28], which makes use of DBpedia to create a map of candidate entities for each term found and to disambiguate the term. In turn, the Topic Extraction Approach produces a set of entities that represent the main subjects of the text. For this task, the approach proposed in [38] was adapted, which makes use of the DBpedia category graph to identify the entities with the greatest relation to the text.

The process of automatic semantic annotation was performed with each data source combination with both annotation approaches. The results were measured using the recall and TopN measures. The TopN is measured as follows: considering a document with a total of N_r manual annotations not ranked for a given video and that were associated with N_k correct annotations by the algorithm, and let $rank_i$ be the position of the i^{th} correct annotation of the response set, thus the TopN is defined as the Equation (1). A constant $\alpha = 0.8$ was adopted to adjust the penalty associated with the correct annotation position in the response. The TopN is used to verify not only whether the algorithm returned correct results, but also how close these results are to the top positions.

$$TopN = \frac{1}{N_k} \sum_{i=1}^{N_k} \frac{\alpha^{rank_i}}{\sum_{j=1}^{rank_i} \alpha^j}. \tag{1}$$

A dataset with manually annotated videos from Videoaula@RNP was created. The test dataset has 39 videos in Portuguese about areas such as computer science, statistics, chemistry and physics, with a total duration of approximately 6 hours. These videos were watched by experts invited to accomplish the manual annotation process. Each specialist assigned a DBpedia feature for each subject explicitly spoken during the video, without repetition[12]. There was no restriction on the number of resources for each video that the specialist could assign. During the process of creating the dataset, it was verified how expensive the manual annotation process is. For every 1 hour of video the experts took an average of 4 hours of manual labor, totaling approximately 24 hours to write down the entire base. We have evaluated the following data sources to choose which are the best data sources for semantic annotation:

Metadata: The texts included by the user, which includes the title, abstract and the keywords extracted from Videoaula@RNP.

Summary: Each video lecture has a summary that describes the topics that will be addressed throughout the lesson.

Speech Recognition: The audio was extracted and the automatic audio transcription was generated using the techniques discussed in Section 3.2.1.

Subtitle: If available, subtitles can be used instead of the automatic transcription. However, subtitles are a manual transcription adapted to better suit the reading. Subtitles are not always present in video repositories due to the high cost of production. This data source has been inserted into the experiments to simulate a speech recognition process with optimal word error rate.

Text Recognition: Text is often present in video lectures, recorded during the slide show, inserted in post-production or in video-related PDF files. OCR algorithms can be used to extract text from video frames.

Table 2 shows the results. Subtitles and speech recognition results demonstrate that the subtitle generates a low TopN, especially in the Entity Linking approach, because it generates a very large set of text and many words that were annotated were not among the entities of the video. In this case, the use of the summary or metadata is more appropriate. In the Topic Extraction approach, the combination of subtitles and transcription generates a high TopN because the Topic Extraction approach is more influenced by the frequency of words in the text. In the Entity Linking approach, the lack of the subtitles can be suppressed by using

[12] `https://github.com/ufjf-dcc/LAPIC1-benchmark`

another data source for higher recall. Like the Topic Extraction approach, there is a satisfactory recall when combining OCR, subtitles and automatic transcriptions.

Source	*Entity Linking*		*Topic Extraction*	
	Recall	TopN	Recall	TopN
M	0.214	0.102	0.071	0.076
M + O	0.611	0.041	0.286	0.132
M + Sb	0.838	0.019	0.410	0.171
M + Sm	0.304	**0.102**	0.132	0.118
M + T	0.614	0.019	0.287	0.160
M + O + Sb	0.838	0.013	0.432	0.334
M + O + Sm	0.630	0.041	0.291	0.129
M + O + T	0.713	0.017	0.351	0.165
M + Sb + Sm	0.838	0.019	0.410	0.185
M + Sb + T	0.838	0.017	0.387	0.162
M + Sm + T	0.656	0.020	0.305	0.161
M + O + Sb + Sm	0.838	0.013	0.432	**0.372**
M + O + Sb + T	0.838	0.012	0.454	0.337
M + O + Sm + T	**0.720**	0.017	**0.356**	**0.162**
M + Sb + Sm + T	0.838	0.016	0.387	0.172
M + O + Sb + Sm + T	0.838	0.012	**0.454**	**0.355**
O	0.568	0.042	0.281	0.109
O + Sb	0.838	0.013	0.432	0.303
O + Sm	0.587	0.042	0.285	0.120
O + T	0.688	0.017	0.347	0.145
O + Sb + Sm	0.838	0.013	0.432	0.362
O + Sb + T	0.838	0.012	0.454	0.327
O + Sm + T	**0.694**	**0.017**	**0.356**	**0.145**
O + Sb + Sm + T	0.838	0.012	0.454	0.344
Sb	**0.838**	**0.020**	0.387	0.156
Sb + Sm	0.838	0.019	0.387	0.194
Sb + T	0.838	0.017	0.387	0.151
Sb + Sm + T	0.838	0.017	0.387	0.172
Sm	0.166	**0.175**	0.098	0.098
Sm + T	**0.590**	0.019	**0.285**	**0.145**
T	**0.531**	**0.017**	**0.264**	**0.145**

Table 2. Recall and TopN using Entity Linking and Topic Extraction approaches. Read M as Metadata, O as OCR, Sb as Subtitle, Sm as Summary, and T as Transcription.

It is worth mentioning that any evaluation study is subject to the quality of the test data. Although the dataset used has been established by experts, it is possible that terms that have been annotated correctly by both approaches were not found on the dataset. The test dataset was created to evaluate how similar the result of the semantic annotation approaches was from manual annotations. Although other analyses can be performed to verify the accuracy of the experiments, the test dataset is adequate for the evaluation that was proposed.

It was possible to see how distinct sources of information can improve the semantic annotation process of educational videos, associating new information that was not previously present in these repositories. The new information represents the video content and would help to understand how the repository is semantically structured. Considering that manually created subtitles are not widely present in videos, automatic transcriptions are used as the main input for our automatic semantic annotation process.

3.2.3 Knowledge Graph

The issues in searching for a specific content as well as the lack of knowledge of the administrators about contents within the repository appear due to the increase of the videos in the repository and a low quality of the tags filled by the users who are disseminating the video. In this type of scenario, even if someone succeed in an initial search for some type of content, it is very difficult to find related content without having to perform a new search. On the other hand, several methods can be used to allow the user to browse the contents of the repository after the initial search.

Some authors propose the use of knowledge bases to help identify similarity between video lectures and other types of videos. In this case, it is common to use text associated with the video, such as titles, abstracts and other metadata, as well as captions or tags filled by users. For example, in [4] the authors manually explore the Wikipedia categories to find the best categories according to the user-created tags and video titles. Next, the Wikipedia categories are used to improve the video categorization.

After the Context Association task, each video is associated with a set of DBpedia resources, i.e., URIs that identify instances in the DBpedia ontology. We use the DBpedia graph of resources and categories in order to identify related videos through a similarity function. The encyclopedic content of DBpedia is suitable for calculating similarity in educational content repositories by having good coverage in the main teaching topics. The Wikipedia corpus covers different fields of knowledge and the organization of its category graph enables the linking of concepts belonging to different domains [19].

The Algorithm 1 is used to calculate the similarity of two videos and to create a relationship among the most similar videos. Let the video v_i be defined as an n-tuple $v_i = \langle r_1, r_2, \ldots, r_n \rangle$ where r_j are DBpedia resources, $j \in [1..n]$. The list C of categories of v_i is the union of the direct categories of each resource $r \in v_i$, the broader (α) and the more specific (β) categories of each direct category of r. We say that the videos v_i and v_j will be related if $sim\left(C_{v_i}, C_{v_j}\right) > \omega$, where $\omega \in [0, 1]$ is a predefined constant.

The similarity between two videos can be calculated as a generic function sim. The Sorensen-Dice coefficient calculation was used as similarity function. This

Algorithm 1: Algorithm for relation prediction

Input : ω, α, β
Output: Set R of related videos

begin
  $R \leftarrow \emptyset$
  for *each video* $v_i \in$ *repository* **do**
    $C_{v_i} \leftarrow \emptyset$
    for *each* $r \in v_i$ **do**
      $c \leftarrow$ getDBpediaCategories(r)
      $C_{v_i} \leftarrow C_{v_i} \bigcup^{\alpha,\beta} c$
    end
  end
  for *each videos* $v_i, v_j \in$ *repository* **do**
    $List_{v_i} = \emptyset$
    if $sim\left(C_{v_i}, C_{v_j}\right) > \omega$ **then**
      append($List_{v_i}, v_j$)
    end
    $R \leftarrow R \bigcup List_{v_i}$
  end
  return R
end

method can be seen in the formula below where Ω represents the percentage of related categories between two videos v_i and v_j.

$$\Omega = \frac{2(|C_{v_i}| \bigcap |C_{v_j}|)}{|C_{v_i}| + |C_{v_j}|}.$$

The relationships between videos, resources and categories are represented in a simplified way by the flowchart in Figure 2. Videos are linked to DBpedia resources by the Contextual Association step, and these resources are associated with categories in the DBpedia graph. Therefore, the similarity is calculated not only by the number of resources that the videos share, but by the number of categories associated with those resources that are gathered after a walk in the graph.

A test dataset was created with manual relationships defined by experts in the area of Exact Sciences in order to evaluate the approach. This version of the dataset is available to other researchers[13] and contains the same set of videos used in the previous section. The experts were free to define relationships between videos. An expert might consider that a video is related by addressing exactly the same topic of a video, but another expert might relate videos considering that they contain complementary information. Altogether, 211 relationships were defined manually,

[13] `https://github.com/ufjf-dcc/LAPIC1-benchmark`

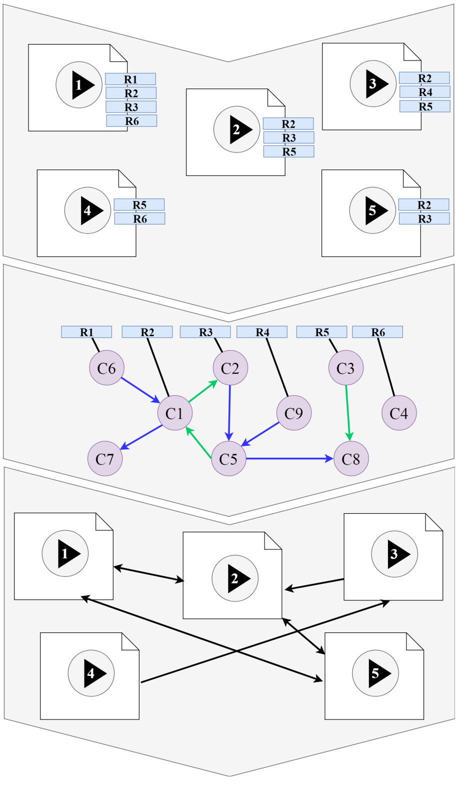


Figure 2. Similarity flowchart

with a mean of 5 relationships per video. There is no video without relations. There are 10 videos with only 1 relationship, and 17 videos with at least 5 relationships.

The approach is expected to find new relationships beyond the existing ones. In the experiments, some combinations of α and β parameters of the algorithm were tested in order to retrieve different levels of DBpedia category information in each of the experiments and to analyze how the amount of information influences the evaluation metrics.

Figure 3 presents the recall and TopN (Y-axis) for each video (or test) in dataset (X-axis). The solid line represents recall values and the dashed line represents TopN values for each experiment. To identify the proportion of videos with good and bad results in each experiment, the videos were sorted by TopN in descending order. A desirable but intangible algorithm that always finds the correct videos would have its constant curves at 1. Figures 3 a), 3 b) and 3 c) show that the recall increases as more information are processed by the algorithm. For instance, Figure 3 a) shows that 9 videos reached a maximum recall (videos numbered from 1 to 9), while Figure 3 b) shows that the maximum recall was reached in 20 videos. It is verified that walking in broader (α) and more specific (β) categories produces high recall. The recall does not change with greater depths according to the obtained results. Although the use of the categories can help in a high TopN, increasing the depth does not imply in higher TopN. By increasing the depth, more common categories of videos will be used, and it will be more difficult to rank the result correctly through the number of categories in common.

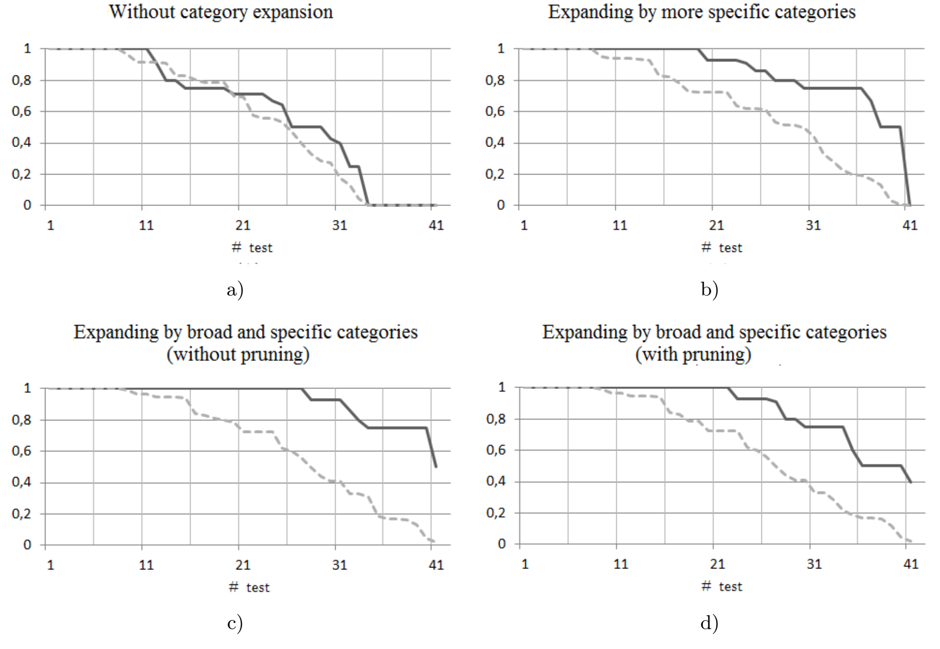


Figure 3. Recall (solid line) and TopN (dashed line) for each experiment

The best configuration found for the parameters was $\alpha = 1$ and $\beta = 1$ (Figure 3 c)), resulting in 32 videos with a recall of approximately 0.9. According to the TopN, the algorithm was able to return the correct videos in the first results in more than half of the dataset. It is also possible to verify that the TopN follows the trend of generating better results as more information is processed by the algorithm. The use of specific categories (Figure 3 b)) presents a better result in relation to the non-expansion approach (Figure 3 a)). Since the list of related videos is ranked for each video, a threshold can be used to limit the set of videos in the result set. Figure 3 d) shows the results of the third experiment retrieving only videos that have at least 10 categories in common. This pruning method did not influence considerably the result of the algorithm. The mean TopN increased from 0.67732 to 0.688215 and the mean recall reduced from 0.93120 to 0.87715.

Figure 4 presents the dispersion of the recall and TopN for the best experiment ($\alpha = \beta = 1$). Each point represents the recall and TopN values of a specific test.

The number of points is equal to the number of tests in Figure 3 c) (X-axis). It can be observed that in the same recall value, the TopN values can vary by up to 20 percentage points. There is a trend in Figure 4, where the higher the recall, the better the ranking of the videos.

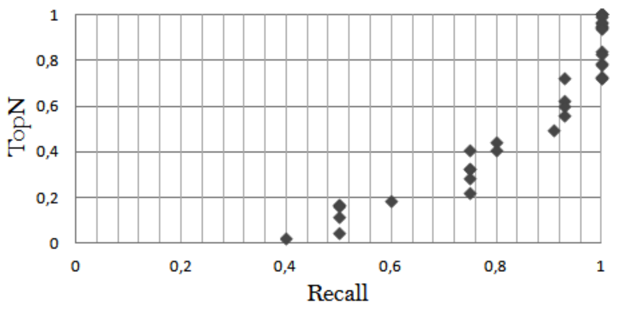


Figure 4. Dispersion of the recall and TopN in the experiment with pruning

Although the results of the experiments were satisfactory, it is important to analyze the false positives. The dataset was created manually by experts following personal criteria to determine which videos should be related. As a result, the test dataset contains videos with few relationships or videos related to others that do not have any resources in common. Thus, the test dataset presents relationships and resources that are not skewed by some set of information or selection methods. Taking into account these particularities, our experiments showed that some false positive relationships that have a large number of categories in common are, in fact, relationships that are absent on the test dataset. That is, the algorithm is able to identify relationships that are not always easily identifiable by a person. Take as an example the video identified as "fis2tempcalor", which addresses concepts of temperature and heat. This video was manually related only to the video identified as "fis2cap18-part2". The algorithm, in turn, related 15 videos with "fis2tempcalor", among them videos about physics and chemistry that address concepts indirectly related to temperature and heat.

The experiments presented in these three sections were carried out with the purpose of demonstrating the feasibility of the proposal. In the following section, we discuss how the approach can be applied to find the main topics in a real video lectures repository.

4 RESULTS AND DISCUSSIONS

In this section, we discuss how the proposed framework can be used in a real video repository. This dataset is composed of 93 randomly selected video lectures from VideoAula@RNP (about 11 % of the repository at the time of the experiment), totaling 3 604 minutes of videos. This repository was created by RNP to make available video lectures produced by associated educational institutions from Brazil.

The videos were transcribed with our ASR system. In the semantic annotation process, each video received up to 5 semantic annotations. In our experiments, the top 5 ranked annotations are enough to generate metadata with high precision. The topic extraction approach was chosen for the Context Association because of the results discussed in Section 3.2.2. Then the knowledge graph was created with $\alpha = 1$ and $\beta = 1$ and no threshold (Section 3.2.3). As a result, each vertex is linked to all others and the edges represent the degree of relationship of the videos (vertices). Figure 5 shows an example of the undirected complete graph. Blue vertices represent a video and green vertices are the categories of the video. The categories related to the videos were extracted from the DBpedia resources automatically annotated in each video and the final graph does not contain the DBpedia resources. The video v3 has a strong relationship with the video v2 because of the number of categories in common. On the other hand, the video v3 shares a few categories with the video v4 and the edge between them has a low value.

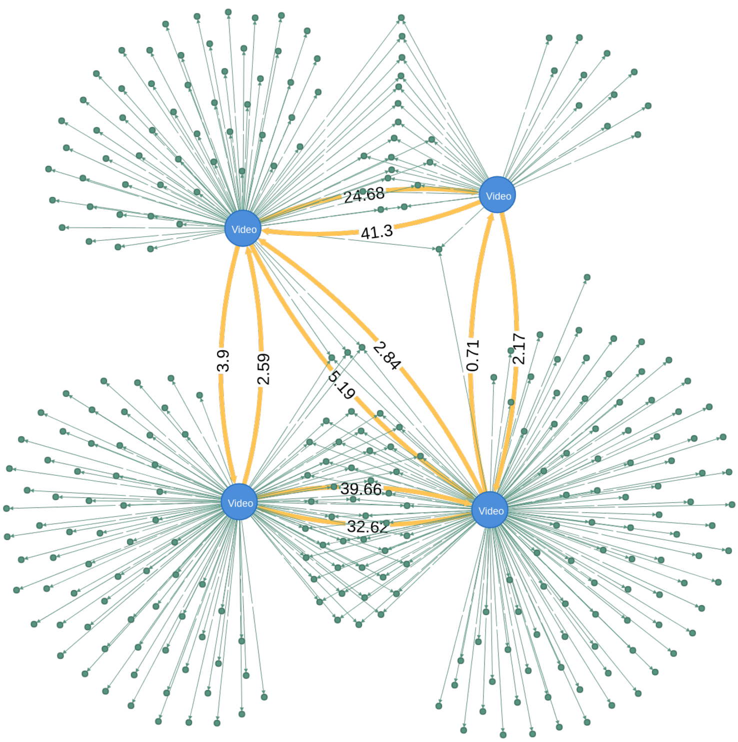


Figure 5. Video relations subgraph. Blue vertices represent videos and green vertices represent categories. Yellow edges link two videos and are weighted. Green edges are unweighted and link one video to one category.

The graph is used to analyze what kinds of content has been produced by users of the repository. The first step consists of identifying communities in that knowledge graph. In this case, communities are clusters of videos that are densely connected internally, that is, groups of videos that share a large number of categories. Therefore, by identifying these communities, we are finding groups of videos that potentially address related subjects.

For this process, the graph was submitted to the Label Propagation Algorithm (LPA), an algorithm for community detection on graphs networks that works by exploring the neighborhoods between the vertices [37]. The algorithm uses network structure alone as its guide, and does not require a pre-defined objective function or prior information about the communities. The algorithm sets a unique label for each vertex. At every iteration of propagation, each vertex updates its label to the one that the maximum numbers of its neighbors belong to. Ties are broken uniformly and randomly. The algorithm reaches convergence when each vertex has the majority label of its neighbors. Figure 6 shows the groups of videos identified by the Label Propagation Algorithm. The graph contains 22 groups of videos with very close subjects. The groups are identified by different colors. The vertices representing categories were removed from the graph for a better view.

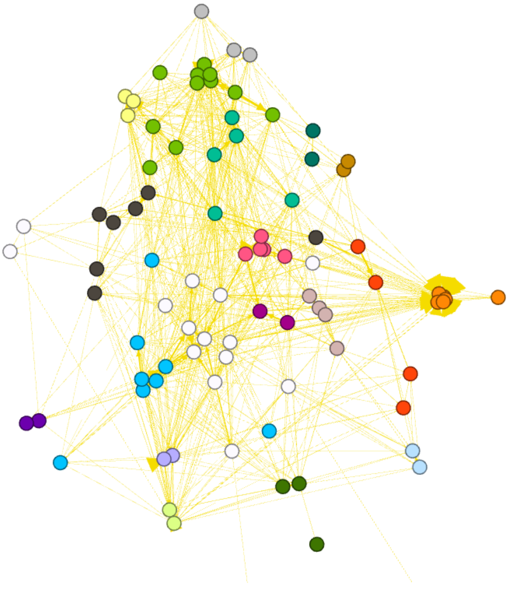


Figure 6. Knowledge graph after the label propagation algorithm

In the second step, we have performed an analysis in the generated groups. For this task, we perform a new search in the graph of categories from DBpedia. Given a set of input resources T, we return a percentage of contribution of each Wikipedia main topic in T. The set of resources T of the group c contains all DBpedia resources of each video automatically annotated of each video in group c. Wikipedia main topics can be seen as end vertices in the DBpedia category graph.

To understand the main topics addressed in each group from the first step, we performed a search in the graph, passing as input the set of resources and walking towards the top of the graph by all the shortest paths between the categories of each resource and the main topics, as proposed in [27]. As a result, a fingerprint is created for each group. A fingerprint is a vector of weights where each dimension represents the weight of that DBpedia main topic. Nowadays, the number of main topics in DBpedia is 19. As an example, the categories used as input for groups 4, 6 and 17 are presented in Table 3. The groups contain 5, 16, and 7 videos, respectively. With the more frequent categories of each group, it is possible to see the difference between the subjects of each group. Group 4 addresses genetics and chemistry topics while Group 6 addresses topics related to sociology, philosophy and law. Group 17 presents subjects focused on artificial intelligence and cryptography.

Group		
4	6	17
Inorganic carbon compounds	Social epistemology	Polynomial-time problems
Alcohols	Social philosophy	Artificial intelligence
Persistent organic pollutants	Philosophy of education	Turing tests
Genetics by type of organism	Theories of law	Cryptographic hardware
DNA repair	Rights	Computer security software
Mitochondrial genetics		Internet fraud
		Internet search algorithms

Table 3. Example of input resources of groups 4, 6, and 17

The results of this step can be seen in Figure 7. The topics of technology, society, geography, culture, and history are present in most of the videos. Religion, life, law, and arts are an example of topics that are not addressed in this videos. Some groups have very similar fingerprints, such as Groups 11 and 15. Although the groups contain videos on history, they are distinct groups because of the difference of the semantic annotations in each group. In other words, the groups encompass videos about history, but the groups have few direct categories in common, addressing distinct subjects in this area of knowledge. In fact, Group 11 contains two videos on political theories and Group 15 contains four videos on wars of independence.

It is possible to see the relationship between Tables 3 and 7. For example, when analyzed, the categories for the Group 6 presented in Table 3 are found to be focused in sociology, philosophy and law. In Figure 7, the Group 6 has a greater weight for the topics of philosophy, presenting 35,65 % of relationships with philosophy and 14.78 % with society. The same occurs in the other groups: in the Group 4 the categories encompass biology and chemistry and the main topics classifications are matter (50 %), health (18.75 %) and nature (12.5 %), for the Group 17 the categories are focused on artificial intelligence and cryptography. The main topics are sciences and technologies (36.95 %) and mathematics (27.17 %).

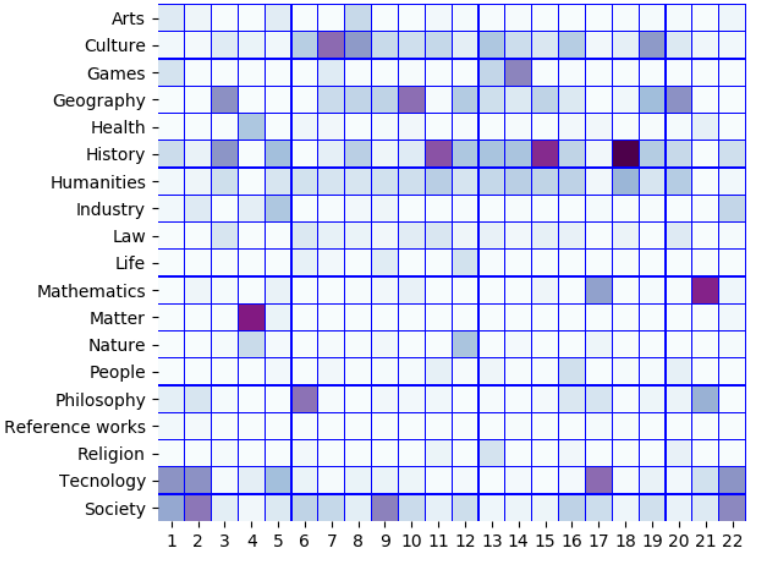


Figure 7. Fingerprint of the groups. The X-axis contains the group numbers and the y-axis contains the Wikipedia main topics.

It is possible to understand the information of the groups automatically, and it can be used to recommend to the user a set of videos that are related in a higher level. Although some topics are classified out of line, as is the case of Group 17 that presents weight in culture (2.17 %), nature (3.2 %) and philosophy (9.78 %), a threshold can be used to filter the results by discarding the lower values.

5 CONCLUDING REMARKS

In this paper, we proposed a framework for knowledge discovering in video lectures repositories. This framework is composed of three main stages: content processing through ASR, context association using semantic annotation and the construction of a knowledge graph through a walk in the DBpedia ontology and similarity calculations. Each part of the framework was evaluated separately and, in the end, the final result of the complete process was used to demonstrate the applicability of the framework in a real repository of video lectures.

As the main contribution of this work, we highlight the applicability of our proposal in video lectures repositories where there is a lack of metadata to describe the videos. As explained throughout the text, this lack of metadata hampers indexing systems, which makes search and recommendation in these repositories very ineffective. Our proposal can automatically extract knowledge from video lectures and can improve several applications in large repositories, such as searching, recommendation, and advertising systems. The advances in automatic speech recognition systems [8, 2, 23] allow developers to easily reproduce these results using well-known ASR tools. The extracted metadata can be used for indexing or clustering, which are everyday tasks in information retrieval and recommendation systems. Our experiments have shown that discovering knowledge allows determining the subjects and relationships of video lectures. Other contributions were the analyses made at each stage of the framework, where it was possible to raise the main points, challenges and solutions for the development of our proposal.

Future work includes analyzing the scope of other knowledge bases in the framework. Other knowledge bases make it possible to find different, more specific resources if a domain base is used. We also intend to study a cross-domain approach with multiple knowledge bases simultaneously. Finally, we also envisage the creation of an interface for navigation in the information found in the repository.

6 ACKNOWLEDGEMENTS

We thank RNP (Brazilian National Education and Research Network) and CAPES (Coordination for the Improvement of Higher Education Personnel) for the financing of this work.

REFERENCES

[1] Asghar, M. N.—Hussain, F.—Manton, R.: Video Indexing: A Survey. International Journal of Computer and Information Technology, Vol. 3, 2014, No. 1, pp. 148–169.

[2] Bahar, P.—Bieschke, T.—Ney, H.: A Comparative Study on End-to-End Speech to Text Translation. IEEE Automatic Speech Recognition and Understanding Workshop (ASRU), Sentosa, Singapore, 2019. arXiv preprint arXiv:1911.08870.

[3] Che, X.—Luo, S.—Yang, H.—Meinel, C.: Automatic Lecture Subtitle Generation and How It Helps. 2017 IEEE 17th International Conference on Advanced Learning Technologies (ICALT 2017), 2017, pp. 34–38, doi: 10.1109/ICALT.2017.11.

[4] Chen, Z.—Cao, J.—Song, Y.—Zhang, Y.—Li, J.: Web Video Categorization Based on Wikipedia Categories and Content-Duplicated Open Resources. Proceedings of the 18th ACM International Conference on Multimedia (MM '10), 2010, pp. 1107–1110, doi: 10.1145/1873951.1874162.

[5] Dasiopoulou, S.—Giannakidou, E.—Litos, G.—Malasioti, P.—Kompatsiaris, Y.: A Survey of Semantic Image and Video Annotation Tools. In: Paliouras, G., Spyropoulos, C. D., Tsatsaronis, G. (Eds.): Knowledge-Driven Multimedia Information Extraction and Ontology Evolution. Springer, Berlin, Heidelberg, Lecture Notes in Computer Science, Vol. 6050, 2011, pp. 196–239, doi: 10.1007/978-3-642-20795-2_8.

[6] Davis, D.—Chen, G.—Hauff, C.—Houben, G.-J.: Gauging Mooc Learners' Adherence to the Designed Learning Path. Proceedings of the 9th International Conference on Educational Data Mining (EDM), 2016, pp. 54–61.

[7] De Medio, C.—Limongelli, C.—Sciarrone, F.—Temperini, M.: MoodleREC: A Recommendation System for Creating Courses Using the Moodle E-Learning Platform. Computers in Human Behavior, Vol. 104, 2020, pp. 106–168, doi: 10.1016/j.chb.2019.106168.

[8] Drugman, T.—Pylkkönen, J.—Kneser, R.: Active and Semi-Supervised Learning in ASR: Benefits on the Acoustic and Language Models. Interspeech 2016, ISCA, 2016, pp. 2318–2322, arXiv Preprint arXiv:1903.02852, doi: 10.21437/Interspeech.2016-1382.

[9] Farrell, R. G.—Liburd, S. D.—Thomas, J. C.: Dynamic Assembly of Learning Objects. Proceedings of the 13th International World Wide Web Conference on Alternate Track Papers and Posters, 2004, ACM, pp. 162–169, doi: 10.1145/1013367.1013394.

[10] Franzoni, V.—Tasso, S.—Pallottelli, S.—Perri, D.: Sharing Linkable Learning Objects with the Use of Metadata and a Taxonomy Assistant for Categorization. In: Misra, S. et al. (Eds.): Computational Science and Its Applications – ICCSA 2019. Springer, Cham, Lecture Notes in Computer Science, Vol. 11620, 2019, pp. 336–348, doi: 10.1007/978-3-030-24296-1_28.

[11] Galanopoulos, D.—Mezaris, V.: Temporal Lecture Video Fragmentation Using Word Embeddings. In: Kompatsiaris, I., Huet, B., Mezaris, V., Gurrin, C., Cheng, W. H., Vrochidis, S. (Eds.): MultiMedia Modeling (MMM 2019). Springer, Cham, Lecture Notes in Computer Science, Vol. 11296, 2019, pp. 254–265, doi: 10.1007/978-3-030-05716-9_21.

[12] Gaona-García, P. A.—Martin-Moncunill, D.—Montenegro-Marin, C. E.: Trends and Challenges of Visual Search Interfaces in Digital Libraries and Repositories. The Electronic Library, Vol. 35, 2017, No. 1, pp. 69–98, doi: 10.1108/EL-03-2015-0046.

[13] González-Pérez, L. I.—Ramírez-Montoya, M.-S.—García-Peñalvo, F. J.: User Experience in Institutional Repositories: A Systematic Literature Review. International Journal of Human Capital and Information Technology Professionals (IJHCITP), Vol. 9, 2018, No. 1, pp. 70–86, doi: 10.4018/IJHCITP.2018010105.

[14] Gruhn, R. E.—Minker, W.—Nakamura, S.: Statistical Pronunciation Modeling for Non-Native Speech Processing. Springer Science and Business Media, 2011, doi: 10.1007/978-3-642-19586-0.

[15] Gupta, Y.—Saini, A.—Saxena, A. K.: A New Fuzzy Logic Based Ranking Function for Efficient Information Retrieval System. Expert Systems with Applications, Vol. 42, 2015, No. 3, pp. 1223–1234, doi: 10.1016/j.eswa.2014.09.009.

[16] Gurbuz, F.: Students' Views on Distance Learning in Turkey: An Example of Anadolu University Open Education Faculty. Turkish Online Journal of Distance Education, Vol. 15, 2014, No. 2, pp. 239–250, doi: 10.17718/tojde.54964.

[17] Habibian, A.—Mensink, T.—Snoek, C. G. M.: Discovering Semantic Vocabularies for Cross-Media Retrieval. Proceedings of the 5th ACM Interna-

tional Conference on Multimedia Retrieval (ICMR '15), 2015, pp. 131–138, doi: 10.1145/2671188.2749403.

[18] Hadian, H.—Sameti, H.—Povey, D.—Khudanpur, S.: Flat-Start Single-Stage Discriminatively Trained HMM-Based Models for ASR. IEEE/ACM Transactions on Audio, Speech, and Language Processing, Vol. 26, 2018, pp. 1949–1961, doi: 10.1109/TASLP.2018.2848701.

[19] Halavais, A.—Lackaff, D.: An Analysis of Topical Coverage of Wikipedia. Journal of Computer-Mediated Communication, Vol. 13, 2008, No. 2, pp. 429–440, doi: 10.1111/j.1083-6101.2008.00403.x.

[20] Harasim, L.: Shift Happens: Online Education as a New Paradigm in Learning. The Internet and Higher Education, Vol. 3, 2000, No. 1-2, pp. 41–61, doi: 10.1016/S1096-7516(00)00032-4.

[21] Hauptmann, A. G.—Jin, R.—Ng, T. D.: Video Retrieval Using Speech and Image Information. Storage and Retrieval for Media Databases 2003, Proceedings of the SPIE, Vol. 5021, 2003, pp. 148–160, doi: 10.1117/12.479747.

[22] Heath, T.—Bizer, C.: Linked Data: Evolving the Web into a Global Data Space. Synthesis Lectures on the Semantic Web: Theory and Technology, Vol. 1, 2011, No. 1, pp. 1–136, doi: 10.2200/S00334ED1V01Y201102WBE001.

[23] Herchonvicz, A. L.—Franco, C. R.—Jasinski, M. G.: A Comparison of Cloud-Based Speech Recognition Engines. Proceedings of the Computer on the Beach, 2019, pp. 366–375.

[24] Hinton, G.—Deng, L.—Yu, D.—Dahl, G. E.—Mohamed, A.-R.—Jaitly, N.—Senior, A.—Vanhoucke, V.—Nguyen, P.—Sainath, T. N.—Kingsbury, B.: Deep Neural Networks for Acoustic Modeling in Speech Recognition: The Shared Views of Four Research Groups. IEEE Signal Processing Magazine, Vol. 29, 2012, No. 6, pp. 82–97, doi: 10.1109/MSP.2012.2205597.

[25] Kota, B. U.—Davila, K.—Stone, A.—Setlur, S.—Govindaraju, V.: Generalized Framework for Summarization of Fixed-Camera Lecture Videos by Detecting and Binarizing Handwritten Content. International Journal on Document Analysis and Recognition (IJDAR), Vol. 22, 2019, pp. 221–233, doi: 10.1007/s10032-019-00327-y.

[26] Lei, W.—Qing, F.—Zhou, J.: Improved Personalized Recommendation Based on Causal Association Rule and Collaborative Filtering. International Journal of Distance Education Technologies (IJDET), Vol. 14, 2016, No. 3, pp. 21–33, doi: 10.4018/IJDET.2016070102.

[27] Medeiros, J. F.—Nunes, B. P.—Siqueira, S. W. M.—Leme, L. A. P. P.: TagTheWeb: Using Wikipedia Categories to Automatically Categorize Resources on the Web. In: Gangemi, A. et al. (Ed.): The Semantic Web: ESWC 2018 Satellite Events (ESWC 2018). Springer, Cham, Lecture Notes in Computer Science, Vol. 11155. 2018, pp. 153–157, doi: 10.1007/978-3-319-98192-5_29.

[28] Mendes, P. N.—Jakob, M.—García-Silva, A.—Bizer, C.: DBpedia Spotlight: Shedding Light on the Web of Documents. Proceedings of the 7th International Conference on Semantic Systems (I-Semantics '11), ACM, 2011, pp. 1–8, doi: 10.1145/2063518.2063519.

[29] Mohan, P.—Brooks, C.: Learning Objects on the Semantic Web. Proceedings of the 3rd IEEE International Conference on Advanced Learning Technologies, 2003, pp. 195–199, doi: 10.1109/ICALT.2003.1215055.

[30] Neves, D. E.—Brandão, W. C.—Ishitani, L.: Automatic Content Recommendation and Aggregation According to SCORM. Informatics in Education, Vol. 16, 2017, No. 2, pp. 225–256, doi: 10.15388/infedu.2017.12.

[31] Ney, H.—Essen, U.: On Smoothing Techniques for Bigram-Based Natural Language Modelling. 1991 International Conference on Acoustics, Speech, and Signal Processing (ICASSP '91), IEEE, 1991, pp. 825–828, doi: 10.1109/ICASSP.1991.150464.

[32] Oliveira, A. L. C.—Silva, E. S.—Macedo, H. T.—Matos, L. N.: Brazilian Portuguese Speech-Driven Answering System. Proceedings of the 6th Euro American Conference on Telematics and Mation Systems (EATIS '12), ACM, 2012, pp. 277–284, doi: 10.1145/2261605.2261647.

[33] Omheni, N.—Kalboussi, A.—Mazhoud, O.—Kacem, A. H.: Recognition of Learner's Personality Traits Through Digital Annotations in Distance Learning. International Journal of Distance Education Technologies (IJDET), Vol. 15, 2017, No. 1, pp. 28–51, doi: 10.4018/IJDET.2017010103.

[34] Piedra, N.—Chicaiza, J. A.—López, J.—Tovar, E.: An Architecture Based on Linked Data Technologies for the Integration and Reuse of OER in MOOCs Context. Open Praxis, Vol. 6, 2014, No. 2, pp. 171–187, doi: 10.5944/openpraxis.6.2.122.

[35] Purarjomandlangrudi, A.—Chen, D.—Nguyen, A.: Investigating the Drivers of Student Interaction and Engagement in Online Courses: A Study of State-of-the-Art. Informatics in Education, Vol. 15, 2016, No. 2, pp. 269–286, doi: 10.15388/infedu.2016.14.

[36] Qazi, A.—Goudar, R. H.: Emerging Trends in Reducing Semantic Gap Towards Multimedia Access: A Comprehensive Survey. Indian Journal of Science and Technology, Vol. 9, 2016, No. 30, doi: 10.17485/ijst/2016/v9i30/99072.

[37] Raghavan, U. N.—Albert, R.—Kumara, S.: Near Linear Time Algorithm to Detect Community Structures in Large-Scale Networks. Physical Review E, Vol. 76, 2007, No. 3, Art. No. 036106, pp. 36–106, doi: 10.1103/PhysRevE.76.036106.

[38] Raimond, Y.—Lowis, C.: Automated Interlinking of Speech Radio Archives. Linked Data on the Web (LDOW 2012) Workshop at WWW 2012, CEUR Workshop Proceedings, Vol. 937, 2012.

[39] Rao, K.—Peng, F.—Sak, H.—Beaufays, F.: Grapheme-to-Phoneme Conversion Using Long Short-Term Memory Recurrent Neural Networks. 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2015, pp. 4225–4229, doi: 10.1109/ICASSP.2015.7178767.

[40] Rojas-Contreras, M.—Portilla-Jaimes, O.: Integration of Learning Objects for Adaptative Learning. IOP Conference Series: Materials Science and Engineering, IOP Publishing, Vol. 519, 2019, Art. No. 012030, doi: 10.1088/1757-899X/519/1/012030.

[41] Sad, S. N.—Goktas, O.—Bayrak, I.: A Comparison of Student Views on Web-Based and Face-to-Face Higher Education. Turkish Online Journal of Distance Education, Vol. 15, 2014, No. 2, pp. 209–226, doi: 10.17718/tojde.02246.

[42] van Merriënboer, J. J. G.—Ayres, P.: Research on Cognitive Load Theory and Its Design Implications for E-Learning. Educational Technology Research and Development, Vol. 53, 2005, No. 3, pp. 5–13, doi: 10.1007/BF02504793.

[43] Vanajakshi, P.—Mathivanan, M.: A Detailed Survey on Large Vocabulary Continuous Speech Recognition Techniques. 2017 International Conference on Computer Communication and Informatics (ICCCI), IEEE, 2017, 7 pp., doi: 10.1109/ICCCI.2017.8117755.

[44] Yang, H.—Meinel, C.: Content Based Lecture Video Retrieval Using Speech and Video Text Information. IEEE Transactions on Learning Technologies, Vol. 7, 2014, No. 2, pp. 142–154, doi: 10.1109/TLT.2014.2307305.

[45] Zhao, B.—Xu, S.—Lin, S.—Luo, X.—Duan, L.: A New Visual Navigation System for Exploring Biomedical Open Educational Resource (OER) Videos. Journal of the American Medical Informatics Association, Vol. 23, 2016, No. e1, pp. e34–e41, doi: 10.1093/jamia/ocv123.

Jorão Gomes Jr. holds a degree in exact sciences from Federal University of Juiz de Fora (UFJF) and is currently pursuing a master's degree from UFJF. He has experience in the area of computer science, with emphasis on information retrieval and natural language processing.

Laura Lima Dias holds her Master's degree in computer science from Federal University of Juiz de Fora (UFJF) (2017). She is researcher at the Laboratory of Applications and Innovation in Computing (LApIC). She has experience in multimedia systems, information retrieval, semantic annotation and informatics in education.

Eduardo Rocha Soares holds his Bachelor's degree in computer science and is currently pursuing his Master's degree from UFJF, also in computer science. He is interested in the area of multimedia systems, with an emphasis on content and natural language processing.

Eduardo Barrere is Professor at Federal University of Juiz de Fora (UFJF), Brazil. He has Ph.D. degree in systems engineering and computing since 2007. He coordinates the LApIC Laboratory (Applications and Innovation in Computing).

Jairo Francisco de Souza is Assistant Professor at the Department of Computer Science, Federal University of the Juiz de Fora (UFJF), Brazil, where he teaches courses in databases and information retrieval. He holds his M.Sc. (2007) from the Federal University of Rio de Janeiro (UFRJ) and his Ph.D. from the Pontifical Catholic University of Rio de Janeiro (PUC-Rio), Brazil, both in computer science area. His research interests include knowledge representation, natural language processing, information integration, and semantic models.

Computing and Informatics, Vol. 38, 2019, 1403–1417, doi: 10.31577/cai_2019_5_1403

FUZZY KNOWLEDGE INFERENCE: QUICKLY ESTIMATE EVIDENCE VIA FORMULA EMBEDDING

Xiao Zhang

School of Mechanical Electronic and Information Engineering
China University of Mining and Technology, Beijing
No. 11 College Road, Haidian District, Beijing China
e-mail: zhangxiao@cumtb.edu.cn

Abstract. Inference on Knowledge Bases (KBs) is an important way to construct more complete KBs and answer KB questions. Inference can be viewed as a process from evidence to conclusion following specific formulas. Traditional methods usually search on the KB to collect evidence, which cannot apply to large-scale KBs, because the running time of searching increases radically as the scale of KBs increases. What is worse, evidence cannot be found if one fact in it is missing, which may result in the failure of inference. To this end, we propose a fuzzy method of estimating evidence, which replaces searching by estimating the existence of evidence by constructing formula embeddings, and then we merge these estimations into a probabilistic model to infer conclusions. This method can apply to large-scale KBs, because estimating evidence is very fast and is irrelevant to the KB scale. Estimating evidence can also be viewed as fuzzy matching, so this method can handle the situation where facts are missing. We evaluate this method on the knowledge base completion task, and it achieves a better performance than state-of-the-art methods and has a shorter running time.

Keywords: Fuzzy logic, fast inference, knowledge base completion, formula mining

1 INTRODUCTION

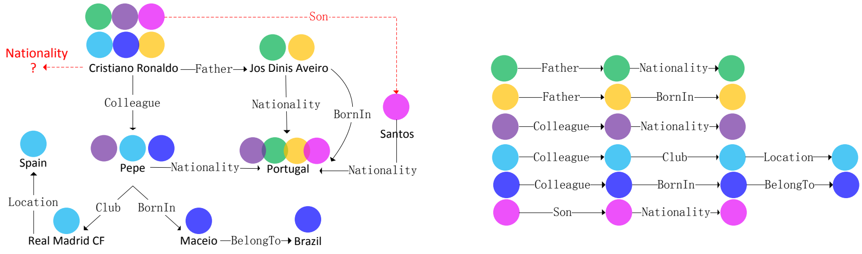
The inference is a process from evidence to a conclusion. A typical inference on KBs is usually to predict the missing element in a tuple, e.g., *Relation (Head Entity, ?)* or *Relation (?, Tail Entity)*, so the inference is an important way to complete KBs and answer KB questions. If we view a KB as direct graphs (Figure 1 a)), inference essentially means to take usage of the graph structures to predict missing links, e.g. *Nationality(Cristiano Ronaldo*[1]*, ?)*. To each specific type of queries, we can extract several frequent structures from graphs as a priori knowledge, and they are called formulas. For example, for *Nationality* queries, $\mathit{Father}(x_1, x_2) \wedge \mathit{Nationality}(x_2, x_3)$ is a frequent structure which has a probability of supporting $\mathit{Nationality}(x_1, x_3)$, and it is called as a formula, where x_i denotes an entity variable. Some other formulas are shown in Figure 1 b).

Traditional inference methods usually search on the KB to collect formula instances as evidence, e.g., we can find $\mathit{Cristiano\ Ronaldo} \xrightarrow{\mathit{Father}} \mathit{Jose\ Dinis\ Aveiro} \xrightarrow{\mathit{Nationality}} \mathit{Portugal}$, to support *Nationality(Cristiano Ronaldo, Portugal)*. Searching evidence cannot apply to large-scale KBs, because its computation complexity is $O(n^l)$, where n is the average degree of nodes and l is the maximal length of formula, and it may take a long time when the graph is large or dense. Another drawback of searching is that its matching condition is too strict. When one or two facts of evidence are missing in the KB, the evidence cannot be found by searching method, which may result in incorrect inference result. For example, *Son(Cristiano Ronaldo, Santos)* is missing in Figure 1 a), so the evidence, $\mathit{Cristiano\ Ronaldo} \xrightarrow{\mathit{Son}} \mathit{Santos} \xrightarrow{\mathit{Nationality}} \mathit{Portugal}$, cannot be found by searching, and further result in the invalidation of the formula, $\mathit{Son}(x_1, x_2) \wedge \mathit{Nationality}(x_2, x_3) \Rightarrow \mathit{Nationality}(x_1, x_3)$.

To accelerate the inference computation process, knowledge graph embedding methods are used in inference on KBs, such as TransE [3], TransH [15] and TransR [10]. TransE is translating embedding. TransH is translating embedding on hyperplanes. Trans means performing translation in relation-specific entity space. PTransE is translating embedding for relation paths. Their basic intuition is to represent entities and relations in KBs as low-rank real vectors, and almost all of them expect that entities in one fact are close in a space specific to the relation. However, these embedding-based methods lack the explicit logic constrains and enough evidence, which makes these methods to be more like modeling KBs than performing inference. Therefore, they are prone to be damaged by unexpected noises and lead to unsatisfying results. Compositional training (COMP) [6] and PTransE [9] have a preliminary attempt at merging paths into KB embedding model, but they have not solved the above problems. COMP uses paths to improve learning KB embeddings rather than to infer missing facts, so its performance is still unsatisfied. PTransE still needs to search paths on KBs, so its running time is as long as the traditional inference methods.

We consider that collecting evidence to infer the conclusion is a good tradition, but searching on KBs is not necessary. KB embeddings provide a possibility to estimate evidence without searching. We exploit the embedding strategy and propose an approach which can realize quick fuzzy inference. This method represents

[1] Cristiano Ronaldo is a Portuguese professional footballer of the Juventus F.C.



a) b)

Figure 1. An example of inference on the KB: a) is the knowledge graph, and the query is what is the nationality of Cristiano Ronaldo, where Son(Cristiano Ronaldo, Santos) is a missing fact; b) is a set of formula operators, which are embedded and used to estimate whether this evidence is occurring between Cristiano Ronaldo and one nation.

formulas as computable operators by using pre-trained KB embeddings, and these formula operators are used to measure the distances from holding evidence. For example, for the formula $\mathit{Father}(x_1, x_2) \wedge \mathit{Nationality}(x_2, x_3) \Rightarrow \mathit{Nationality}(x_1, x_3)$, we use embeddings of *Father* and *Nationality* to represent the formula operator, and then it is applied on *Cristiano Ronaldo* and *Portugal* to estimate the existence of the evidence, $\mathit{Cristiano\ Ronaldo} \xrightarrow{\mathit{Father}} \mathit{Somebody} \xrightarrow{\mathit{Nationality}} \mathit{Portugal}$. This method treats the evidence as a whole and only cares about whether it exists. After that, we merge the probabilities of evidences obtained from estimation into the MLN framework and infer the probability of *Nationality(Cristiano Ronaldo, Portugal)* is true. To realize evidence fuzzy matching and speed up the estimation, we ignore middle variables, e.g., the middle variable, *Jose Dinis Aveiro*, is replaced with *Somebody*. Therefore, this method is insensitive to missing facts and can find evidence under difficult situations where facts are missing in the evidence. When estimate one evidence, the computation complexity of this method is $O(1)$, and paths with any length can be estimated by one step simple calculation. Therefore, our approach can apply to large-scale KBs.

2 RELATED WORK

In general, according to the process of inference on KB, there are three types of approaches:

1. probabilistic logic inference;
2. knowledge graph embedding;
3. formula embedding.

Especially, the third type can be viewed as the combination of logic and embedding.

2.1 Probabilistic Logic Inference

This type of methods focused on mining frequent substructure on the KB as evidence, and it models the relationship between evidence and the query in different ways. MLN [14], PSL [5] and PRA [8] are all typical probabilistic logic models. These methods need searching or performing random walks to collecting evidence, and then they use a probabilistic model to combine evidence to infer conclusions. We take MLN as the example to explain, and its manner of combining evidence is similar to our approach.

Markov Logic Network (MLN) can be viewed as a probabilistic extension of first-order logic by attaching weights to formulas. Especially, for its discriminative version, higher weight indicates greater reward to the query that satisfies the formula. To predict the query Y, it mines a set of formulas F_Y and counts these formula groundings on the KB, which is the typical practice of search-based inferences. Then MLN calculates Y's conditional probability as follows:

$$P_w(Y = y|X = x) = \frac{1}{Z_y}\exp\left(\sum_{f_i \in F_Y} w_i n_i(x, y)\right) \tag{1}$$

where $n_i(x, y)$ denotes the number of true groundings of formula f_i and w_i is f_i's weight. $Z_y = \sum_{y' \in Y} exp(\sum_{f_i \in F} w_i n_i(x, y'))$ is the normalizing term.

2.2 Knowledge Graph Embedding

Many knowledge graph embedding models are proposed in recent years, such as RESCAL [13], SE [4], SME[2], LFM [7], TransE [3], TransH [15], TransR [10]. Almost all embedding-based models embed entities into a relatively low (e.g., 50) dimensional embedding vector space R^k while representing relations in different ways. These models expect that correlative elements of the KB are close in the embedding space, so they can be used to complete KBs. To predict the query $r(h, t)$, most of the embedding-based models calculate the similarity between E_h and E_t under E_r, noted as $f_s(E_h, E_t, E_r)$, where E_h, E_t and E_r represent the embeddings of h, t and r, respectively. This type of methods can be used to learning KB embeddings before estimate evidence in our approach, but they do not represent explicit evidence. We employ TransE as an example to explain how the embedding model performs inference, and we use TransE to learn KB embeddings in our experiments.

TransE represents relations as translations in the entity vector space R^k, and assumes that if $r(h, t)$ holds, then the embedding of the tail entity t should be close to the embedding of the head entity h plus some vector that depends on the relationship r. TransE's similarity function is:

$$f_{r(h,t)} = -\|E_h + E_r - E_t\|_2^2. \tag{2}$$

TransE employs a margin-based ranking loss to learn the KB embeddings. TransE assumes that $f_{r(h,t)}$ is larger than other $f_{r(h,t')}$ or $f_{r(h',t)}$, where $r(h,t)$ exists in the KB but $r(h,t^{'})$ or $r(h^{'},t)$ does not and designs its loss function $\mathcal{L}$ as follows:

$$\mathcal{L} = \sum_{f_s \in KB} \sum_{f_s' \in KB'} [\gamma + f_s^{'} - f_s]_+ \tag{3}$$

where KB and $KB^{'}$ represent a true fact set and a false fact set according the KB, respectively. The symbol $[\ldots]_+$ means the final results always bigger than 0. When the number in [] is less than zero, the final result is 0. When the number in [] is bigger than 0, the final result is the number.

2.3 Formula Embedding

Embedding formulas is latest research subject, and researchers focus on combining logic and embedding which is also our purpose. Compositional training (COMP) [6], PTransE [9], SePLi [17] and RNN model [12] are all representative models, and they are related to our method of embedding paths.

COMP wants to learn KB embeddings by using facts and paths, simultaneously. COMP represents paths in two ways:

1. RESCAL based: a path is represented as a matrix, which equals the produce of relations in it;
2. TransE based: a path is represented as a vector, which equals the sum of relations in it.

COMP constructs a dataset of paths by performing random walks on the KB, and then the path dataset and the original KB are used to learn KB embeddings.

PTransE has a similar method to represent paths, which is also based on TransE. PTransE designs its loss function by combining both original query loss and path loss, and designs its loss function as follows:

$$L = L(h,r,t) + \frac{1}{Z} \sum_{p \in P(h,t)} R(p|h,t) L(p,r) \tag{4}$$

where $L(h,r,t)$ is the query loss and the second term is the path loss. $P(h,t)$ is the set of paths from h to t, and $R(p|h,t)$ is a kind of probability of path p. PTransE uses the similarity between the path p and the relation r to define the path loss. However, $L(p,r)$ is irrelevant with entities in the query, so it is like the weight of p rather than its loss.

3 OUR APPROACH

We first describe our fuzzy inference model. We gradually propose three types of evidence under three assumptions and explain why we finally employ formula operators as evidence. Then we describe the method of estimating evidence by KB embeddings.

3.1 Evidence-Based Inference Model

A typical inference method usually designs a model to combine evidence to infer the conclusion, and they define, collect and filter evidence according to their own models. How to define and acquire evidence is of crucial importance for correct inference. When we infer a query, *Relation(Head Entity, Tail Entity)*, evidence can be all visible information which is related to the query. For the above example in Figure 1, the part of knowledge graph connected to *Cristiano Ronaldo* or *Portugal* is the whole evidence. We propose the first assumption as follows:

Assumption 1. A connected subgraph including both Head Entity and Tail Entity is the whole evidence for predicting any relation between these two entities. If another subgraph is also evidence, there must be missing facts between these two subgraphs.

The assumption can be proved easily, because any irrelevant entity or fact in the KB is useless for predicting the query. When another subgraph affects the inference result, We can always add a link under a relation type to connect entities in these two subgraph. The assumption give the first glance of of the importance of missing facts. Under Assumption 1, an inference model is shown as:

$$f_{r(h,t),s} = W(r(h,t), s) \cdot (1 - D(s)) \tag{5}$$

where $r(h,t)$ is the query, s is the subgraph evidence, and $W(r(h,t), s)$ is the weight of inferring $r(h,t)$ from s. D function represents the distance from finding evidence, and it is defined by the specific method of acquiring evidence. When the method fully finds s, $D(s) = 0$, and when the method never find s, $D(s) = 1$.

Although the connected subgraph contains complete and original information, the whole structure is too sparse and difficult to used by learning models. To solve the problem of sparse feature, We split the evidence graph into independent parts. The most intuitive and convenient substructure is path (where path is a general path which may contain reverse edges), so we make the following assumption.

Assumption 2. All paths connecting Head Entity and Tail Entity are the whole evidence for predicting any relation between these two entities. If an unconnected path is evidence, it always can link Head and Tail Entity by adding facts to it.

This assumption splits the connected subgraph in Assumption 1 into several paths between head entity and tail entity, and these paths are used in the subsequent inference model independently. Similar to Assumption 1, when an unconnected path may have a contribution on inferring the query, we can make it connected by adding links under one specific relation. For example, the unconnected path

Santos $\xrightarrow{Nationality}$ *Portugal* in Figure 1 does not connect to *Cristiano Ronaldo*, and then we can add a link from *Cristiano Ronaldo* to *Santos* under *Son* relation to make it connected. Such situations of missing facts are common in KBs. Under Assumption 2, we also give an inference model, as:

$$f_{r(h,t),P} = \frac{1}{Z} \sum_{p_i \in P} w_i \cdot (1 - D(p_i)) \tag{6}$$

where P is the set of paths connecting h and t, w_i is the weight of p_i, and $Z = \sum_{p_i \in P} w_i$ is the normalizing constant. $D(p_i)$ is the distance from finding p_i, and the range of $D(p_i)$ is also [0,1]. For traditional search-based methods, $D(p_i)$ in Equation (6) has only two values: 1 or 0. When a fact is missing from p_i, $D(p_i)$ is always 1. Therefore, search-based methods cannot handle all paths in Assumption 2.

To hold paths with missing facts, we find that estimating whether some paths exist between entities is easier than searching them exactly. Therefore, we extract relation sequences from paths, and treat them as formula operators to estimate paths between entities. For example, we can obtain the relation sequence *Colleague* $\rightarrow$ *Club* $\rightarrow$ *Location* as an operator from the path *Cristiano Ronaldo* $\xrightarrow{Colleague}$ *Pepe* $\xrightarrow{Club}$ *Real Madrid CF* $\xrightarrow{Location}$ *Spain*, and we can use the formula operator to estimate whether such paths exist between any other two entities. Therefore, we propose the third assumption.

Assumption 3. All types of formula operators existing between Head Entity and Tail Entity are the whole evidence for predicting any relation between these two entities.

This assumption takes a formula operator as a whole by ignoring middle entities and only cares about whether a type of paths occurs. According to Assumption 2, if one middle entity is useful, we can always construct another path which is from Head Entity to Tail Entity and passes the middle entity. There are two advantages of Assumption 3:

1. we can use one distance to estimate the existence of one path type and ignore the number of paths.
2. Paths under the same relation sequence share weights, which reduces the feature sparsity.

We change the inference model in Equation (6) as follows:

$$f_{r(h,t),F_r} = \frac{1}{Z} \sum_{f_{r_i} \in F_r} w_{r_i} \cdot (1 - D(f_{r_i})) \tag{7}$$

where F_r is the set of formula operators for inferring relation r, and $D(f_{r_i})$ is the distance from formula operator f_{r_i} occurring. A formula operator is not a path but a set of paths, and the distance $D(f_{r_i})$ is used to estimate whether there is at least one path under this formula operator.

These three assumptions are layers of the progress. They define three forms of evidence: connected subgraph, path, and formula operator, and these three types of evidence should contain all information required by corresponding inference model. Inference models treat these evidence as features and employ a linear model to merge evidence by attaching weights to them. The simple linear model requires the representation of evidence containing dependency and interactions between elements in KBs, and the evidence representation decides how to acquire evidence and how to define the distance function D.

3.2 Estimate Evidence

We propose our approach under Assumption 3 and represent formula operators as computable real vectors to estimate evidence and quickly calculate distance function $D(f_{r_i})$ in Equation (7). To make formula operators computable, we propose to represent them by KB embeddings.

Embedding means representing each entity in KB as a low-dimension numeric vector, and different dimensions of the vector may implicitly represent different aspects of an entity. Relations in KB usually have relevant representations, such as vectors, matrixes and tensors. Entities interact under a specific relation by performing arithmetical operations between entity embeddings and relation's representation.

To make embedding of a formula operator contain its relations' information and share information with other related formula operators, we exploit the idea of TransE [3], COMB [6] and PTransE [9]. TransE represents relations as translations in the entity vector space R^k, and assumes $E_h + E_r = E_t$ when $r(h,t)$ holds. COMB and PTransE treat a path as a normal relation, and have $E_h + E_p = E_t$ when $p(h,t)$ holds. For a path $h \xrightarrow{r1} x \xrightarrow{r2} t$, we can get $E_h + E_{r_1} = E_x$ and $E_x + E_{r_2} = E_t$. We rewrite the second equation as $E_x = E_t - E_{r_2}$ and use it to eliminate the middle variable x, and then we get $E_h + E_{r_1} + E_{r_2} = E_t$. $E_{r_1} + E_{r_2}$ is naturally used as the representation of the formula operator $\rightarrow r1 \rightarrow r2 \rightarrow$. More generally, we define one formula operator as $E_f = \sum_{r_i \in p} E_{r_i}$.

We also treat a formula operator as a translation. When a formula operator f occurs between h and t, we expect E_h to be close to E_t under E_f. Therefore, we define the $D(f_{r_i})$ in Equation (7) as $D(f_{r_i}) = |E_h + E_{f_{r_i}} - E_t|$, and use it to estimate the probability of existing at least one path under f_{r_i} between h and t. For example, if we want to know whether there is at least one path $\xrightarrow{Father} x \xrightarrow{Nationality}$ between *Cristiano Ronaldo* and *Portugal*, we just calculate distance by $\|E_{C.Ronaldo} + E_{father} + E_{nationality} - E_{Portugal}\|_2^2$ and ignore who *Cristiano Ronaldo*'s father is. When the grounding path *Cristiano Ronaldo* $\xrightarrow{Father}$ *José Dinis Aveiro* $\xrightarrow{Nationality}$ *Portugal* exists in the KB, the distance should be close to 0.

We prove the correctness of the estimating algorithm. When we find a path $p = e1 \xrightarrow{r_1} e2 \xrightarrow{r_2} \ldots \xrightarrow{r_n} e_{n+1}$ by search on the KB graph, if all facts $r_i(e_i, e_{i+1}) \in p$

exist in the KB, we can declare the path p exist. This criterion can be viewed as a necessary and sufficient condition, and we add up each fact's distance to existence as the path distance.

$$\begin{aligned} D(p) &= \sum_{r_i(e_i,e_{i+1})\in p} \|E_{e_i} + E_{r_i} - E_{e_{i+1}}\| \\ &\geq \| \sum_{r_i(e_i,e_{i+1})\in p} E_{e_i} + E_{r_i} - E_{e_{i+1}}\| \\ &= \|E_{e_1} + \sum_{r_i\in f} E_{r_i} - E_{e_{i+1}}\| \\ &= \|E_{e_1} + E_f - E_{e_{i+1}}\| \end{aligned} \tag{8}$$

where $\|\cdot\|$ can be any norm function as a fact distance. According to triangle inequality for norms, the sum of norms is greater than or equal to the norm of the sum, so we can get the second line. The middle entities can be eliminated and there are only relations left as the third line, and the sequence of relations can be viewed as a formula operator. Coincidentally, we have $E_f = \sum_{r_i\in f} E_{r_i}$ and finally get the forth line. The path p's real distance $D(p)$ must be greater than or equal to the distance of the formula operator $D(f) = \|E_{e_1} + E_f - E_{e_{i+1}}\|$. When $D(f) = 0$, we can get all facts $\|E_{e_i} + E_{r_i} - E_{e_{i+1}}\| = 0$, what indicates that the path p exists. Therefore, it is reasonable that we employ $D(f) = \|E_{e_1} + E_p - E_{e_{i+1}}\|$ to estimate whether there is at least one path under formula operator f.

Dataset	Relation	Entity	Train	Valid	Test
WN18	18	40 943	141 442	5 000	5 000
FB15K	1 345	14 951	483 142	50 000	59 071

Table 1. Statistics of WN18 and FB15K

Before estimating formula operators, this approach needs firstly mining a set of formula operators for each relation type. We employ the Depth-First-Search (DFS) algorithm to traverse the KB graph several times to count frequent relation sequence, and limit the maximum length of paths. Then we simply rank the formula operators by occurrence number, and choose top-K as the set of formula operators. How to mine formula operators is not a focus of this paper, so we avoid missing useful formula operators just by choosing a large K.

3.3 Objective Formalization

We rewrite Equation (7) as the score function for a triplet $r(h,t)$, as:

$$f_{r(h,t),F^r} = 1 - \frac{1}{Z} \sum_{f_i^r \in F_r} w_i^r \cdot \|E_h + E_{f_i^r} - E_t\|_2^2 \tag{9}$$

	Dataset	WN18				FB15K			
	Metric	Mean Rank		Hits@10(%)		Mean Rank		Hits@10(%)	
		Raw	Filt	Raw	Filt	Raw	Filt	Raw	Filt
2.a	INS*(MLN)	329	319	55.5	66.33	242	226	49.2	60.3
2.b	RESCAL	1 180	1 163	37.2	52.8	828	683	28.4	44.1
	TransE	263	251	75.4	89.2	243	125	34.9	47.1
	TransH	401	388	73.0	82.3	212	87	45.7	64.4
	TransR	238	225	79.8	92.0	**198**	77	48.2	**68.7**
2.c	COMB	504	491	78.1	90.7	212	92	39.9	52.6
2.d	FIEE	**133**	**127**	**93.5**	**96.9**	221	**76**	**49.8**	68.6

Table 2. Knowledge base completion results

where w_i^r measures the correlation between formula operator f_i^r and the query relation r, and $w_i^r > 0$ represents positive correlation to predicting the query and $w_i^r < 0$ represents negative correlation. Especially, to take advantage of the implicit relationship between entities as TransE does, we treat r itself as a special formula operator. In Equation (9), only w is parameter and needs to be learnt by training model, while we use embedding E pre-trained by TransE and never change them. We employ a margin-based rank loss to training model, as:

$$\mathcal{L} = \sum_{f_s \in KB} \sum_{f_s' \in KB'} [\gamma + \sum_{p_i^r \in P_r} w_i^r \cdot (D_{f_s'}(p_i^r) - D_{f_s}(p_i^r))]_+ \tag{10}$$

where $D_{f_s}(p_i^r)$ and $D_{f_s'}(p_i^r)$ represent path distances of true and false queries, respectively.

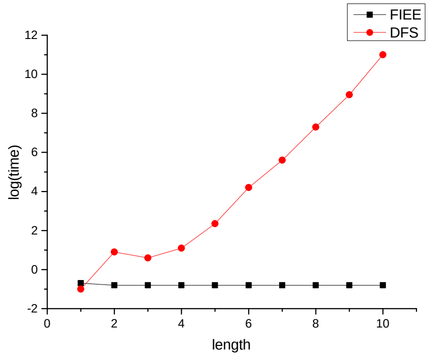
4 EXPERIMENTS AND ANALYSIS

We have compared our approach with several state-of-the-art methods on KB completion (KBC) task. We predict the missing h or t for a fact $r(h,t)$ in the test set. The detail evaluation method is to replace t in $r(h,t)$ by all entities in the KB to form left testing set Q_{left}, and methods need to rank the right answer at the top of the list. Similarly, we produce the right testing set Q_{right}. We perform the experiments on both WN18 and FB15K datasets which were subsets sampled from WordNet[11] and Freebase [1], respectively, and Table 1 shows statistics of them. We report the mean of those true answer ranks and the Hits@10[2] under both 'raw' and 'filter' as TransE does.

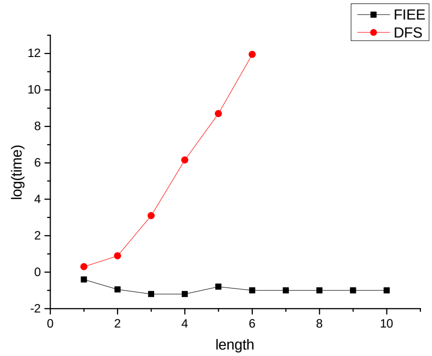
For comparison, we employ 3 types of methods as baselines:

1. search-based method: MLN [14];
2. embedding-based methods: RESCAL [13], TransE [3], TransH [15], TransR [10];
3. embedding and path method: COMB [6].

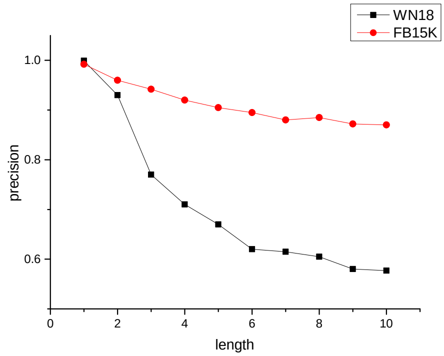
[2] The proportion of correct entities ranked in the top 10.



a) Runtime on WN18



b) Runtime on FB15K



c) Precision

Figure 2. The running time and performance of estimating evidence

We employ INS* [16] as the implement of MLN model. For embedding methods, we use reported results directly since the evaluation datasets are identical. We also implement COMB algorithm, and generate its path query dataset randomly.

We call our approach as FIEE (Fuzzy Inference by Estimate Evidence). For our approach, we select different parameters for two datasets. We use stochastic gradient descent (SGD) to learn embeddings and weights, and we employ validate set to select parameters. We select the learning rate λ among {0.001, 1e-4, 1e-5, 1e-6}, the margin γ among {0.125, 0.25, 0.5, 1}, the dimensionality of entity and relation k among {50,100}, the L_2 regularization coefficient among {0.1, 1, 10} and the maximum length of paths l_{max} among {2,3,4,5,6,7} for WN18 and {2,3,4} for FB15K. Optimal configurations are: $\lambda = 0.001$, $\gamma = 0.5$, $k = 50$, $L_2 = 10$ and $l_{max} = 4$ for WN18; $\lambda = 0.001$, $\gamma = 1.0$, $k = 100$, $L_2 = 0.1$ and $l_{max} = 3$ for FB15K.

4.1 Our Method vs. State-of-the-Art Methods

Table 2 shows the KBC results on both WN18 and FB15K, and we can obtain the following observations.

1. FIEE achieves good performances on both WN18 and FB15K. For WN18, our approach outperforms all state-of-the-art methods on all metrics. It indicates that our approach is effective for knowledge inference.
2. Comparing FIEE with INS in Table 2 a), FIEE outperforms INS on all metrics. It indicates our fuzzy inference is more robust to missing facts than search-based inference. Noise paths can weaken INS seriously and its performance gets worse as the number of candidates grows, which is described in [16]. On the other hand, FIEE treats all entities in the KB as candidates, which implies fuzzy inference can weaken the negative effect of noise paths.
3. Comparing FIEE with embedding-based methods in Table 2 b), FIEE outperforms them on metrics except Mean Rank(raw) and Hits@10(filt) on FB15K dataset. It indicates that introducing explicit logic constrains into the embedding model can improve the KBC performance, and it shows that knowledge graph embedding models have limited inference ability.
4. Comparing FIEE with COMB in Table 2 c), FIEE outperforms COMB. It indicates that paths used as evidence have more effect than used as additional training data, though both of them have the same way of representing paths. Comparing TransE, COMB has no obvious advantage, which also indicates that COMB cannot utilize structure information completely.
5. The best performance of our approach occurs when the maximum path length is 4 for WN18 and 3 for FB15K. This implies there would be more noise when the path length increases in both estimating evidence and the KBC task.
6. Our approach's performance on FB15k is not as good as on WN18. We think the reason may be that FB15K has much more relation types than WN18, which means there would be huge size of formula operators in FB15K. Therefore, it is hard to cover all correlative and meaningful formula operators, and some of ones we selected may be noise.

4.2 Comparison on Running Time

To prove our approach's absolute predominance on running time, we design an experiment to compare the running time of our estimating algorithm and Depth-First-Search algorithm.

We construct a dataset which contains 10 000 various true or false (1:1) path instances with length from 1 to 10, and run FIEE 100 times on it to estimate paths. As comparison, DFS algorithm searches same path instances 100 times, and we compare their running times.

Figures 2 a) and 2 b) show methods' running times on WN18 and FB15K, respectively. We show logarithms of running time and find the running time of DFS increasing exponentially with the growth of path length. Especially on FB15K, DFS cannot finish in 72 hours when path length reaches 6. However, our approach's running time almost remains unchanged with path length increasing and always is under 1 second. It shows our approach's victory by great superiority on running time and indicates that our approach is high-efficient for estimating evidence no matter how long it is.

4.3 Estimate Evidence

The performance of estimating formula operators has an important effect on the KBC result, so we take an experiment to evaluate our approach's performance on it. We still employ the dataset in running time experiment and use estimating path existence to approximate formula operators. Figure 2 c) shows the precision on both WN18 and FB15k, and we can obtain the following observations:

1. Our approach achieves a good performance on estimating paths, which proves our approach can replace searching paths on the large-scale graph.
2. The precision is reducing with the growth of path length both in WN18 and FB15k. For WN18, when the length reached 5, the precision was less than 70 %. There are two reasons of this phenomenon:

   (a) there are cascading errors which are mentioned in [6], and the longer the path is, the larger the cascading error is;
   (b) there may be missing facts in paths, and the longer the path is, the more facts are missing.

   The interesting thing is that the first reason has adverse effect on KBC task, while the second reason is reverse. There is an optimum maximum length of paths for each specific KB.
3. The precision of estimating path existence on FB15K is higher than the precision on WN18, but the KBC performance on WN18 is better. It further proved that our approach can handle paths with missing facts, and these paths can improve KBC performance.

5 CONCLUSION AND FUTURE WORK

This paper presents a novel method, called FIEE, to perform fuzzy inference by estimating evidence. FIEE treats relation sequences as computable formula operators to estimate path existence between entities. FIEE never searches on KBs, so its running time is short and it can be applied to large-scale KBs. Estimating evidence can also be viewed as fuzzy matching, so this method can handle the situation where facts are missing. We evaluate our approach on the KBC task, and it achieves good performances on both WN18 and FB15K datasets.

In future, we will explore the following research directions:

1. The precision of estimation is not very high, and we think the reason is that the method of representing formula operator is not good enough, so we need explore better representation of formula operators.
2. In this paper, we still need to get formula types before learning the inference model, and then we want to deal with them simultaneously.

REFERENCES

[1] Bollacker, K.—Evans, C.—Paritosh, P.—Sturge, T.—Taylor, J.: Freebase: A Collaboratively Created Graph Database for Structuring Human Knowledge. Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data (SIGMOD '08), 2008, pp. 1247–1250, doi: 10.1145/1376616.1376746.

[2] Bordes, A.—Glorot, X.—Weston, J.—Bengio, Y.: A Semantic Matching Energy Function for Learning with Multi-Relational Data. Machine Learning, Vol. 94, 2014, No. 2, pp. 233–259, doi: 10.1007/s10994-013-5363-6.

[3] Bordes, A.—Usunier, N.—Garcia-Durán, A.—Weston, J.—Yakhnenko, O.: Translating Embeddings for Modeling Multi-Relational Data. In: Burges, C. J. C. et al. (Eds.): Advances in Neural Information Processing Systems 26 (NIPS 2013), Vol. 2, 2013, pp. 2787–2795.

[4] Bordes, A.—Weston, J.—Collobert, R.—Bengio, Y.: Learning Structured Embeddings of Knowledge Bases. Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence (AAAI '11), 2011, pp. 301–306.

[5] Brocheler, M.—Mihalkova, L.—Getoor, L.: Probabilistic Similarity Logic. Proceedings of the Twenty-Sixth Conference on Uncertainty in Artificial Intelligence (UAI 2010), 2010, pp. 73–82.

[6] Guu, K.—Miller, J.—Liang, P.: Traversing Knowledge Graphs in Vector Space. Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2015, pp. 318–327, doi: 10.18653/v1/D15-1038.

[7] Jenatton, R.—Le Roux, N.—Bordes, A.—Obozinski, G. R.: A Latent Factor Model for Highly Multi-Relational Data. In: Pereira, F. et al. (Eds.): Advances in Neural Information Processing Systems 25 (NIPS 2012), 2012, pp. 3167–3175.

[8] Lao, N.—Mitchell, T.—Cohen, W. W.: Random Walk Inference and Learning in a Large Scale Knowledge Base. Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing (EMNLP), ACL, 2011, pp. 529–539.

[9] Lin, Y.—Liu, Z.—Luan, H.—Sun, M.—Rao, S.—Liu, S.: Modeling Relation Paths for Representation Learning of Knowledge Bases. Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2015, pp. 705–714.

[10] Lin, Y.—Liu, Z.—Sun, M.—Liu, Y.—Zhu, X.: Learning Entity and Relation Embeddings for Knowledge Graph Completion. Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence (AAAI '15), 2015, pp. 2181–2187.

[11] Miller, G. A.: Wordnet: A Lexical Database for English. Communications of the ACM, Vol. 38, 1995, No. 11, pp. 39–41, doi: 10.1145/219717.219748.

[12] Neelakantan, A.—Roth, B.—McCallum, A.: Compositional Vector Space Models for Knowledge Base Completion. Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), 2015, pp. 156–166, doi: 10.3115/v1/P15-1016.

[13] Nickel, M.—Tresp, V.—Kriegel, H.-P.: A Three-Way Model for Collective Learning on Multi-Relational Data. Proceedings of the 28th International Conference on Machine Learning (ICML '11), 2011, pp. 809–816.

[14] Richardson, M.—Domingos, P.: Markov Logic Networks. Machine Learning, Vol. 62, 2006, No. 1-2, pp. 107–136, doi: 10.1007/s10994-006-5833-1.

[15] Wang, Z.—Zhang, J.—Feng, J.—Chen, Z.: Knowledge Graph Embedding by Translating on Hyperplanes. Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence (AAAI '14), 2014, pp. 1112–1119.

[16] Wei, Z.—Zhao, J.—Liu, K.—Qi, Z.—Sun, Z.—Tian, G.: Large-Scale Knowledge Base Completion: Inferring via Grounding Network Sampling over Selected Instances. Proceedings of the 24th ACM International Conference on Information and Knowledge Management (CIKM '15), 2015, pp. 1331–1340, doi: 10.1145/2806416.2806513.

[17] Wu, F.—Song, J.—Yang, Y.—Li, X.—Zhang, Z.—Zhuang, Y.: Structured Embedding via Pairwise Relations and Long-Range Interactions in Knowledge Base. Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence (AAAI '15), 2015, pp. 1663–1670.

Xiao Zhang is a teacher in the China University of Mining and Technology, Beijing. She received her Ph.D. degree in computer software and theory from the Institute of Software, Chinese Academy of Sciences, Beijing in 2018. She received her B.Sc. degree in computer science and technology from the Ji Lin University in 2012. Her research interests include operating systems and distributed systems.

Computing and Informatics, Vol. 38, 2019, 1418–1443, doi: 10.31577/cai_2019_5_1418

GENERALIZED SELECTION METHOD

Jana Lovíšková

Institute of Informatics
Slovak Academy of Sciences
Dúbravská cesta 9
845 07 Bratislava, Slovakia
e-mail: `jana.loviskova@savba.sk`

Daniel Pernecký

Faculty of Electrical Engineering and Information Technology
Slovak University of Technology in Bratislava
Ilkovičova 3
812 19 Bratislava, Slovakia
e-mail: `xperneckyd@stuba.sk`

Abstract. In this paper we introduce new selection method, 3-selection method. This method tries to generalize the most used selection methods in Genetic Algorithms (GA). Our new method involves both proportional and rank-based methods (order-based) and, moreover, it allows scaling of selection pressure with higher precision. This method is based on defining the shape of probability density distribution which is adjustable by parameters of our method. In addition, our method has one more attribute which adds randomness of selection.

Keywords: Selection method, evolutionary algorithm (EA), genetic algorithm (GA), continuous scalability of selection pressure, continuous scalability of randomness

Mathematics Subject Classification 2010: 97R40

1 INTRODUCTION

Evolutionary algorithms as metaheuristic optimization methods stand on few basic ideas. These ideas came from evolutionary principles found in living nature. By the influence of evolution, nature was able to make incredibly specialized and adapted species, by its own. Adaptation process to surrounding conditions could be seen as some kind of optimization. Adaptation process is conditioned by surroundings pressure, which determines the quality-fitness of individuals by life or death. Therefore surrounding is a quality criterion. Individual's quality is taken into account in two aspects. Fitter individual has a higher probability to transmit its good genetic information to offsprings and, moreover, has a higher probability to outlive than less fit individuals. This mechanism is called Darwin's natural selection.

Next key inspiration was already mentioned in information transmission from parents to offsprings. This mechanism consists of few joined principles. One of them is coding individual's properties into form of genetic sequence/string – chromosome, where each part represents one property/variable of problem being solved (optimization task). This string can be then modified by variation operators. Variation operators could be in form of recombination – crossover or mutation like in living nature. Chromosome modification allows to create individuals with new properties. Another main aspect is the existence of more solutions/individuals (population) at the same time. Unlike in classic optimization methods which work with only one solution. The existence of population is necessary for application of evolutionary principles, but, moreover, it allows to search the space of feasible solutions in parallel.

In evolutionary algorithms two main principles are used. Variation (recombination/crossover and mutation) which creates potential solutions and selection which applies the quality criterion. Selection mechanism is one of the key processes in evolutionary algorithms. Selection setup directly influences the speed of optimization process and quality of the solution found. If selection policy is setup to prefer only the best solutions, then it usually leads to the searching process stuck in local optimum also called as premature convergence. On the other hand, if selection strategy which has very low selection intensity is chosen then the time needed to find feasible solution could be too long, and the solution could not be found at all.

In basic EA the selection is applied in two phases. Choosing individuals – parents for reproduction process or other variation operator such as mutation, and choosing which individuals will survive to a next generation – called survivor selection.

1.1 Generational/Steady State EA

The application time and place/order of selection can significantly change the whole algorithm behavior. In the genetic algorithms there exist two modifications of such different usage. The first type and mostly used is a generational model of EA. Generational EA selects a big portion of individuals from the population. Selected individuals make pairs of parents and undergo variation operations. In the next step the survivor selection is used. New population is a joined group of modified individuals and only a few unchanged individuals from the old population.

The second type is a steady-state EA. In each generation only one pair of parents is selected. Parents undergo the variation operations and these modified individuals undergo the survivor selection. The survivor selection in this case decides which individuals in the population will be replaced by new children individuals.

These two types of EA differ only in the usage of the selection, therefore they have different selection schemes. The different selection scheme could be seen as unimportant diference, but it significantly changes algorithm behavior [1].

However, selection is not only used in the survivor and the parents selection. As the field of EA increased new types of algorithms were developed.

As mentioned previously, the selection operator in evolutionary algorithms is very important and has a huge influence on the convergence speed and on the quality of the final solution. Selection by itself is independent on the EA dialect. Whether it is a genetic algorithm, an evolutionary strategy, a genetic programming or other EA dialect, selection method is generally applicable for all types because of its only dependence on the individual's fitness or individual's genotype. The axiom of preferring better individuals on the expense of the worse ones is a base for most of the selection methods and fitness dependence is widely used.

The aim of our work was to develop a new fitness-based selection method, which is an extension and improvement of the already existing fitness-based selection methods. Our selection method involves both proportional and order-based methods and, moreover, it allows scaling of selection pressure with higher precision over the range of its possible values $\langle 0, 1\rangle$. Scaling of selection pressure with so high precision over the range of its possible values is not allowed by any of the fitness-based selection methods up to now. Our method has one more attribute which adds randomness of selection from range $\langle 0, 1\rangle$.

All of these three attributes (generalization) allow our proposed selection method, within a particular tackled task (solution landscape) and within the specific settings of algorithm's decision parameters, to achieve a better solution of the task than other well-known fitness-based selection methods.

We named this new general selection method based on the prescribed form of a probability density distribution as 3-selection method. We discuss it in detail in Section 3.

In addition, we have proposed a new classification of selection methods because we consider the schemes found in the current existing resources as improper and uncomplete. Our proposed classification of selection methods is stated in Section 2.

2 RELATED WORKS

First selection method was proposed by Holland in [2]. This method tried to take fitness of all individuals into account as objectively as possible. So, the logical step was to take individuals' fitnesses to a proportion, with individuals with better fitness having a higher possibility to be chosen than the worse ones. This method imitates a roulette wheel. Roulette wheel is divided into pieces with different sizes. Each size is proportionate to the fitness of each individual where better individuals have bigger portion of the wheel and vice versa. The wheel is then spin and the marked individual is chosen. The roulette as a hazard simulates aspect of surviving hazard-randomness in the living nature.

This method was later modified and extended. The modifications tried to solve the problem of premature convergence. The most used one is a stochastic universal selection (SUS) method, which has, in general, the same outcome as the roulette wheel selection method, but it solves a worse statistical properties of the basic roulette wheel method [3]. Despite the inherent simplicity of the roulette wheel method, it has been recognized that the roulette wheel algorithm does not, in fact, give particularly a good sample of the required distribution. Whenever more than one sample is to be drawn from the distribution, the use of the stochastic universal selection (SUS) algorithm is preferred [4].

Another classic method which works on completely different principles and one of the most used selection method is a tournament selection. It is a very powerful and simple selection algorithm.

Simply the method compares randomly chosen individuals and the better one is chosen. Here the rate of fitness difference or fitness proportion is not considered, but the main role is taken on individuals fitness order.

Comparison on tournament selection and roulette wheel selection and its modifications was made by more authors, for example in [3].

Natural development in EA research area has produced many other selection methods, which tried to solve premature convergence problem or tried to increase the quality of the final solution process by different approaches. Due to the existence of various selection methods based on different principles it is necessary to categorize the selection methods in order to achieve a clear and comprehensive overview.

At present, the categorization of the selection methods has not been clearly defined and there are not many resources where selection methods are strictly classified (to our knowledge there are only two resources). The authors in the first scheme [6] classified selection methods from the historical point of view. Authors in the second scheme [7] categorized selection methods in the base of their operation to "proportional", "ordinal" (or "order-based") and "steady state" categories.

But there exist many other selection methods which work on a different principle. More methods which select individuals based on their differences were developed. This type of selection tries to solve the premature convergence problem, by not using fitness as the selection criterion but as some kind of genotype metrics such as diversity.

Selection methods can be divided into fitness-dependent (classic selection methods), genotype-dependent (selection methods reflecting genotype) and special methods. Fitness-dependent methods can be further divided into proportional and order-based. Category of a special methods contains for example random selection method, which does not belong to any of the other categories, but also methods such as correlative tournament selection and correlative family-based selection.

Two main categories belonging to group of the fitness-based selection are the proportional selection methods and the order-based methods. Proportional selection selects individuals based on their fitness values relative to the fitness of the other individuals in the set population. Developed scaling policies are able to manipulate the fitness proportions distribution in population [5]. Order-based selection strategy also called ranking is based on order/rank of fitness values in population [14, 6, 7]. These methods were developed to overcome problems in proportional selection methods. For proportional methods there is a general disadvantage in cases where a very fit individual could come across among the population and that will significantly increase the selection intensity. In the order-based methods it is not important how big is the difference between individuals, whether it is few times more, or it is just a small difference, the ordering is the same.

With respect to the abovementioned facts we consider the first scheme [6] and the second scheme [7] as improper and uncomplete.

Therefore we proposed already mentioned own classification. In the proposed classification, the selection methods are divided into three main groups.

1. Fitness-based selection methods

   Fitness-dependent methods are divided into proportional and order-based. Proportional selection methods include: Roulette wheel selection (SSR, SSPR) [2, 9], Stochastic universal selection (SUS) [10], Stochastic remainder selection (with replacement, without replacement), and Deterministic sampling. Order-based selection methods include: Elitism, Tournament selection (Binary tournament selection, Larger tournament selection, Boltzmann tournament selection [11, 12]), Linear ranking selection [5, 8, 14, 15], Exponential ranking selection [8], and Truncation selection [8].

2. Genotype-based selection methods

   This group includes the Diversified selection method and methods based on gender-specific selection: Genetic algorithm with chromosome differentiation (GACD) [16], Restricted mating [17, 18, 19, 20], Genetic relatedness-based selection, Fitness uniform selection scheme (FUSS) [21], and Reserve selection [22].

3. Special selection methods

   This group consists of the following methods: Random selection, Correlative tournament selection [13], and Correlative family-based selection [13].

The proposed classification is a summary of selection methods found in literature. Unfortunately, only few of these methods have been seriously analyzed and compared [8, 3]. Some researchers tried to analyze selection methods by some measurable characteristics [3, 23, 8]. The most frequently used measurable characteristics are takeover time and selection pressure [3, 23]. These characteristics can be used to compare selection methods in a certain way and can provide an information why is some setting of evolutionary algorithm better than the other for particular cases. But they can say nothing about which selection method is more suitable than the other in general, because a different setting of selection pressure and takeover time may be appropriate in particular cases. Convergence of the EA depends also on the settings of the other algorithm's decision parameters and the solution landscape is a priori unknown (no free lunch theorem).

3 SELECTION METHOD BASED ON PRESCRIBED PROBABILITY DENSITY FUNCTION (3-SELECTION METHOD)

Selection based on the prescribed form of a probability density distribution arised from the idea to construct a general selection method for genetic algorithms. It represents the original, our suggested selection method for the fitness-based selection methods. If this method is sufficient in general, it must provide a wide scalability of selection rate, from a purely random selection to the elitist behavior.

The main role of selection is to maintain perspective pieces of information contained in the genotype, therefore a selection method must naturally prefer better individuals over the worse ones. Hence each selection method is based on the fact that a fitter individual has a higher probability of selection than a less fit individual. We solve minimization tasks, which means that the most fit (fittest) individual represents the minimum value of the fitness function.

Our method is based on defining the shape of probability density distribution. This shape of selection method must satisfy the criteria (if we assume minimization problem):

$$p(f_1) \leq p(f_2) \leq \ldots \leq p(f_i) \leq \ldots \leq p(f_n)\,; f_1 \geq f_2 \geq f_3 \geq \ldots \geq f_i \geq \ldots \geq f_n \tag{1}$$

where

- $p(f_i)$ is probability of selection of i^{th} individual whose fitness is f_i,
- f_1 is fitness value of the least fit individual,
- f_n is fitness value of the fittest individual.

If we preserve property defined in (1) and we consider that the prescribed probability curve is defined with some function $p(f_i) = F(f_i)$, then we can convert this feature to a specific selection method. The algorithm of a "general" selection is as follows (minimization problems):

1. Normalization of fitness values in the interval $\langle 0, 1\rangle$ by Equation (2) for proportional selection and by Equation (4) in case of order-based selection.
2. Selection of random individual whose normalized fitness is f_{n_i}.
3. If $F(f_{n_i}) \geq r$ then this individual is copied to a group of selected individuals; r is random generated number from the range $(0, 1\rangle$.

4. If the number of selected individuals equals to the number of individuals to be selected then end, else go to step 2.

Standardization of fitness values in the interval $\langle 0,1\rangle$ is performed in our selection from two reasons. First, for the generality of different scales of fitness landscapes. Second, for our prescribed form of the probability density distribution function. Standardization of fitness values is done in such a way that the fittest individual (in our case, with the minimum fitness value) will have the value of a normalized fitness 1, and the least fit individual (in our case, with the maximum fitness value) will have the value of normalized fitness 0. Other individuals will have fitness values between the boundary points of the interval $\langle 0,1\rangle$.

Fitnesses and normalized fitnesses have not the same ordering due to the construction of our prescribed form of the probability density distribution function, which is in form of (5), (6).

$$f_{n_i} = 1 - \frac{(f_i - \min(f))}{(\max(f) - \min(f))} \to f_{n_i} \in \langle 0,1\rangle \,. \tag{2}$$

Taking into account the fact that the selection methods can be generally classified into two main groups, namely a proportional and order-based group, a logical consequence is that if we replace in step 1 of our algorithm the standardization of fitness values for the standardization of order, considering sorted individuals:

$$\operatorname{srt}(f) = f_1 \geq f_2 \geq f_3 \geq f_4 \geq f_i \geq \ldots \geq f_n, \tag{3}$$

$$f_{n_i} = \frac{i}{N} \to f_{n_i} \in (0,1\rangle \,, \tag{4}$$

then we get the order-based selection method.

Our goal was to make the method of selection as general as possible, so it needs to be adjustable from a random selection to an elitist selection. Function F was therefore chosen in the form of (5), (6) where $\varphi \in \langle 0,1\rangle$ is the input parameter which defines selection pressure of the method. In (6), if $\varphi = 1$ we assume that $1^{infinity} = 1$ and numbers less than 1 including zero raised to an infinity tend to 0.

- If $\varphi \leq 0.5$

$$F(f_{n_i}) = f_{n_i}^{2\varphi}, \tag{5}$$

- if $\varphi > 0.5$

$$F(f_{n_i}) = f_{n_i}^{\frac{0.5}{1-\varphi}}. \tag{6}$$

With regard to another important feature of our selection method (adding randomness of selection), at first we clarify how the selection, respectively the feature of selective pressure, behaves in the evolution of the genetic algorithm.

If the selective pressure is high, it is causing fast convergence just by new individuals emerging in the population being created from the best individuals, and therefore the algorithm is searching in the direction of the fastest descent criterion function. This is due to the fact that selection allows manifestation of only this information which manifests itself immediately or in a few generations and from this reason some random change caused by a mutation here has not a big chance to survive and consequently to manifest. When the selective pressure is high, the algorithm otherwise converges quickly, but in the case of hard multimodal functions it results as stuck on local suboptimal solution.

On the other side, when the selective pressure is too low the algorithm has good ability to avoid local extremes but the time of convergence to the solution may be too long. In this case, the information resulting from global mutation have a big chance to survive longer because the selection allows individuals to carry this information to get to the next generation. However, a disadvantage is that many times the unperspective areas of the task are unnecessarily scanned, and it increases the time complexity of the whole algorithm.

A well-functioning selection method should, on the one hand, provide sufficient ability to increase the selective pressure, but, on the other hand, it should provide the possibility of randomness, it means a certain chance for unsuccessful individuals to survive as well. Consequently, there comes the possibility to combine worse and better individuals, and thereby to increase the probability of finding a global solution. We tried to incorporate this idea into the proposed selection method. In case of the probability density distribution shapes (Figure 1) it means shifting the whole curve upwards, and that is caused by the additional randomness (Figure 3). In other words, to a given probability density distribution curve we add a uniform probability density distribution with a certain amplitude σ, where $\sigma \in \langle 0,1\rangle$ is a random parameter. Parameter of selection pressure $\varphi \in \langle 0,1\rangle$. The final formula for function F is:

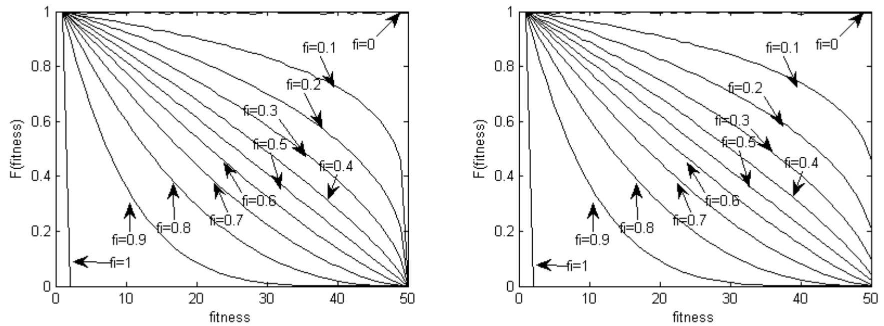
- if $\varphi \leq 0.5$
$$F(f_{n_i}) = (f_{n_i}^{2\varphi} + \sigma)/(1+\sigma), \tag{7}$$

- if $\varphi > 0.5$
$$F\left(f_{n_i}\right) = \left(f_{n_i}^{\frac{0.5}{1-\varphi}} + \sigma\right) / \left(1+\sigma\right). \tag{8}$$

In (8), if $\varphi = 1$ we assume that $1^{infinity} = 1$ and numbers less than 1 including zero raised to an infinity tend to 0.

Probabilistic selection model of 3-selection method with adding randomness of selection:

- if $\varphi \leq 0.5$
$$p\left(X_i\right) = \frac{\left(f_{n_i}^{2\varphi} + \sigma\right) z\left(f_i\right)}{\sum\left(f_{n_i}^{2\varphi} + \sigma\right)}, \tag{9}$$



a) b)

Figure 1. Generated shape of selection probability according to Equations (5) and (6). On the left we can see the proportional selection for different parameters of φ, on the right there is the order-based selection. In this illustrative example, on the x-axis there are 50 fitness values where 1 is the best value and 50 is the worst value. On the y-axis we can see a corresponding selection probability of $F\left(f_{n_i}\right)$ for parameter φ.

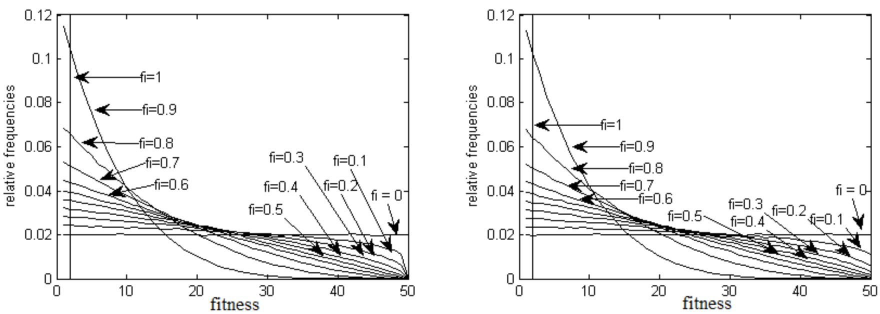


Figure 2. Corresponding relative frequencies for selection curves in Figure 1. On the left side there is the proportional selection method for different φ values, on the right side there is the order-based selection. On the x-axis there are 50 fitness values where 1 is the best value and 50 is the worst value. On the y-axis we can see a corresponding relative frequencies for parameter φ.

- if $\varphi > 0.5$

$$p\left(X_i\right) = \frac{\left(f_{n_i}^{\frac{0.5}{1-\varphi}} + \sigma\right) z\left(f_i\right)}{\sum\left(f_{n_i}^{\frac{0.5}{1-\varphi}} + \sigma\right)}, \tag{10}$$

$z\left(f_i\right)$ is the number of repeating of the i^{th} individual (fitness) in the population.

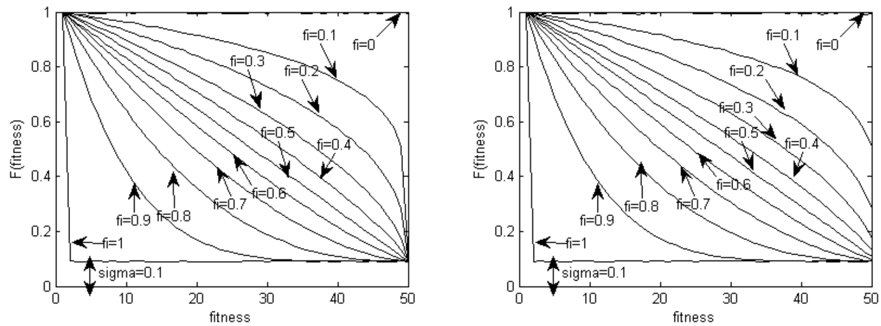


Figure 3. Generated shape of selection probability according to Equations (7) and (8). On the left there is the proportional selection for different parameters of φ, on the right there is the order-based selection. In this illustrative example, on the x-axis there are 50 fitness values where 1 is the best value and 50 is the worst value. On the y-axis we can see a corresponding selection probability of $F\left(f_{n_i}\right)$ for parameter φ.

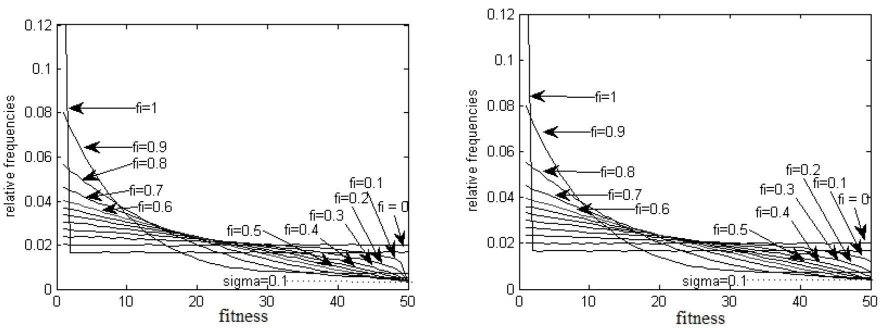


Figure 4. Corresponding relative frequencies for selection curves in Figure 3. On the left side there is the proportional selection method for different φ values, on the right side there is the order-based selection. On the x-axis there are 50 fitness values where 1 is the best value and 50 is the worst value. On the y-axis we can see a corresponding relative frequencies for parameter φ.

A significant advantage resulting from the stated properties of our selection is the ability to continuously change the value of selective pressure (φ) as well as the degree of randomness in the selection (σ).

To clarify the way 3-selection method works we present the algorithm of our proposed 3-selection method (Algorithm 1).

Algorithm 1: 3-selection method

Data: selection rate φ, rand rate σ, pp – proportional ($pp = 1$) or order-based selection ($pp = 2$), pop – input population, fit – fitness vector of individuals in input population, n – required number of selected individuals, norm_fit – normalized fitness

Result: popout – output population, fitout – fitness of individuals in output population

```
initialization popout, fitout;
if φ > 0.9999 then
    φ = 0.9999;
end if
φ = φ * 2;
if pp = 1 then
    if max(fit) − min(fit) ≠ 0 then
        norm_fit = (fit − min(fit))./(max(fit) − min(fit));
        norm_fit = 1 − norm_fit;
    else
        norm_fit = (fit)./(max(fit));
    end if
else
    if pp = 2 then
        sort pop according to descending fitness values;
        sort fit according to descending fitness values;
        nn – number of fitness values in fitness vector;
        norm_fit = (1 : nn) /nn;
    else
    end if
end if
count = 0;
```

4 EXPERIMENTS

The right setup of any evolution algorithm is not an easy task due to many variables needed to be set and our method provides 3 more additional parameters, and that could be seen as a disadvantage. On the other hand, it provides a possibility of very precise setting of a selection.

The influence of 3 new parameters is shown on 6 different GA (Table 1), which differ in the variation operators setup. We chose different GA setup in the meaning of a different exploration and exploitation rate. The results for different combinations of 3-selection method parameters on the 6 different GA setups are compared to the tournament and the SUS selection methods.

```
while n > count do
    j = round to the nearest integer towards infinity
      (random number * population size);
    if φ ≤ 1 then
        if random number * (1 + σ) ≤ σ + (norm_fit(j)^(φ)) then
            count = count + 1;
            save individual j from pop to popout to position given by the
              value of variable count;
            save fitness of individual j from fit to fitout to position given by
              the value of variable count;
        end if
    end if
    if φ > 1 then
        if random number * (1 + σ) ≤ σ + (norm_fit(j)^(0.5/(1−(φ/2)))) then
            count = count + 1;
            save individual j from pop to popout to position given by the
              value of variable count;
            save fitness of individual j from fit to fitout to position given by
              the value of variable count;
        end if
    end if
end while
```

In the experiments for all tested functions we used a simple panmictic GA whose algorithm was:

1. Generate initial population of 50 individuals (chromosomes) – each individual (chromosome) consists of 5 genes and each gene generate randomly from the considered range of values (searching space) for particular test function.
2. Fitness evaluation of new or modified individuals – minimization problem – the most fit (fittest) individual has minimum value of the fitness function.
3. Selection of 3 groups of individuals:
    - Best – one best individual,
    - Old – 15 random selected individuals,
    - Work – 34 individuals selected by 3-selection method.
4. Crossover of the Work individuals.
5. Global mutation of the crossed individuals.
6. Local mutation of the mutated individuals.
7. Merging of groups Best, Old and Work to the new population.

8. If end condition is satisfied then end, else go to step 2.

The termination condition was set as the stagnating convergence of GA for 200 generations with 0 difference of best fitness. This allows to get the best result for a GA setup. The global mutation is a mutation which can modify a gene with a value from the whole searching space. The mutation probability is defined as (number of genes) $*$ (number of individuals) $*$ (mutation rate) $*$ (uniform random number).

For the local mutation an additive mutation was used, which adds a value from a space $\langle -\text{amps}, +\text{amps} \rangle$ to the mutated gene. The *amp* value is a maximal amplitude of the additive mutation, usually taken as a very small part of the searching space.

GA-1: low-div-LOCAL
50 individuals
5 genes
One-point crossover
Uniform global mutation 5 % Local (additive) mutation 5 % Amplitude of local (additive) mutation – 0.1 % from the space range

GA-2: mid-div-LOCAL
50 individuals
5 genes
One-point crossover
Uniform global mutation 20 % Local (additive) mutation 20 % Amplitude of local (additive) mutation – 0.1 % from the space range

GA-3: high-div-LOCAL
50 individuals
5 genes
One-point crossover
Uniform global mutation 50 % Local (additive) mutation 50 % Amplitude of local (additive) mutation – 0.1 % from the space range

GA-4: low-div-GLOBAL
50 individuals
5 genes
One-point crossover
Uniform global mutation 5 % Local (additive) mutation 5 % Amplitude of local (additive) mutation – 10 % from the space range

GA-5: mid-div-GLOBAL
50 individuals
5 genes
One-point crossover
Uniform global mutation 20 % Local (additive) mutation 20 % Amplitude of local (additive) mutation – 10 % from the space range

GA-6: high-div-GLOBAL
50 individuals
5 genes
One-point crossover
Uniform global mutation 50 % Local (additive) mutation 50 % Amplitude of local (additive) mutation – 10 % from the space range

Table 1. Genetic algorithm settings for different exploration and exploitation rate

In the experiments, we used 5 different test functions, namely Eggholder function, Quadratic function, Fnc1 function, Rastrigin function and Sgu2 function. Each of them has a different geometrical landscape and therefore a different degree of difficulty. In addition, each of the test function needs a totally different setup of decision parameters, namely the degree of a selection pressure, the degree of a randomness, the global and the local mutation rate and the amplitude of local mutation. For all of the test functions a common parameter was the number of searching space dimensions and it was set to 5.

The combinations of 3-selection method parameters were made as full factorial of selection φ and random σ parameters with 0.05 step and with both order and proportional type of selection. All of the provided results show the average of 100 runs for every combination.

In our article we provide only the best results (SR, MBF) compared to the results from other applied selection methods – SUS, tournament selection. For each used test function we only state the GA setup in which the best result is found (see Tables 2, 3, 4, 5, 6, 7, 8, 9, 10, 11). The best results of success rate (SR) and mean best fitness (MBF) function are marked bold.

There are 5 292 different combinations of parameters values and GA variation operators setups for each test function. Concretely, 21 selection pressure setting options, 21 randomness setting options, 2 selection methods (proportional, order-based) and 6 different GA setups ($21 * 21 * 2 * 6 = 5\,292$). Our sel3 selection method gradually goes through all possible combinations of its parameter values and is looking for an optimal solution for each combination (therefore, the parameters of the selection method are not pre-set, all of them are passed sequentially).

The most interesting results for each of the 6 different GA setups (due to 5 292 of combinations per test function) compared to results for tournament and SUS selection for each test function we could present due to the large scale of tables in the Annex. As can be seen from these tables (listed in the Annex) for each test function, (for the indicator for SR and also for the indicator for MBF) the best results through the proposed 3-selection method were achieved.

The exception is evaluating the indicator of success rate (SR) for functions Rastrigin and Quadratic (less difficult test functions) – here the best result (100 %) was reached not only using selection method sel3 but also using the conventional standard selection methods SUS and tournament selection. For more difficult test functions (Eggholder, Fnc1, Sgu2) this is no longer true while the best results for the indicator of success rate (SR) and mean best fitness (MBF) are achieved only using the method sel3.

Of course, each of the test functions is, in general, achieving the best results by certain specific settings of variation operators of GA, which directly influence diversity and the level of degree of searching. Thus, in a certain specific (for given function the most suitable or a number of the most suitable) setting(s) of variation operators each of the used and tested selection methods shows better results for given test function in comparison with other settings of variation operators.

If we compare 3-selection method with the proportional type and with the order-based type using our five test functions, we can see that their effectiveness is relatively balanced. Whether it is, in a particular case, more effective to apply the proportional or order-based selection method always depends on the type of the test function and partly also on the setting of variation operators of genetic algorithm.

4.1 Eggholder Function

This test function has a strong multimodal character. The variables of this function are not linearly dependent, and it increases the difficulty of finding a solution. Global extreme of this function is unknown. By now the best reached minimal value of this function for 5 variables of the considered range $\langle -500; 500\rangle$ is $-3\,719.7$.

$$\begin{aligned} f(X) = &\sum_{i=1}^{n-1}\left(-x_i \sin\left(\sqrt{|x_i - (x_{i+1}+47)|}\right)\right) \\ &- \sum_{i=1}^{n-1}(x_{i+1}+47)\sin\left(\sqrt{\left|x_{i+1}+47+\frac{x_i}{2}\right|}\right). \end{aligned} \tag{11}$$

success rate [%] SR -3650	SUS	sel3 proportional	Tournament	**sel3 order-based**
high-div-local	29 %	64 % p-sel = 0.65 p-rand = 0 and p-sel = 0.9 p-rand = 0.3	61 %	**67 %** **p-sel = 0.65** **p-rand** = 0.15

Table 2. Eggholder function – SR

mean best fitness MBF	SUS	sel3 proportional	Tournament	**sel3 order-based**
high-div-local	-3 274.18	-3 515.68 p-sel = 0.65 p-rand = 0.1	-3 502.99	**-3 538.92** **p-sel = 0.65** **p-rand = 0.15**

Table 3. Eggholder function – MBF

4.2 Quadratic Function

Unimodal function. This function has only one extreme, but, for testing purposes, it is very useful. The value of extreme of this function is 0, where $x_i = 0$ for $i = 1, 2, 3, \ldots, 5$; $x_i \in \langle -500; 500\rangle$.

$$f(X) = \sum_{i=1}^{n} x_i^2. \tag{12}$$

success rate [%] SR 0,1	**SUS**	**sel3 proportional**	**Tournament**	**sel3 order-based**
low-div-local	**100 %**	**100 % all parameters p-sel, p-rand**	**100 %**	**100 % all parameters p-sel, p-rand**

Table 4. Quadratic function – SR

mean best fitness MBF	SUS	sel3 proportional	Tournament	**sel3 order-based**
mid-div-local	0.0000126	0.0001259 p-sel = 1 p-rand = 0.15	0.0000653	**0.00000147** **p-sel = 0.95** **p-rand** = 0

Table 5. Quadratic function – MBF

4.3 Fnc1 Function

Fnc1 function is designed so that every position and value of extreme is known. It consists of one declining hyper-space of x^3 and randomly generated Gaussian functions. For this test function, it is characteristic that for finding a solution a high degree of randomness is needed. The randomness rate should be at least 0.45. It seems that for this test function and for the proportional selection (when randomness rate ≥ 0.5), to find the best results, the size of the selection parameter does not matter much. We would like to discuss this phenomenon in more detail in the following article.

$$\sum_{j=1}^{ex} \prod_{k=1}^{dim} \frac{-\sqrt{\frac{s_1(j)}{\Pi}}}{e^{s_2(j)(x(k)-o(k,j))^2}} + \left(\sum_{i=1}^{dim} 0.002 x_i \right)^3 . \tag{13}$$

The parameters s_1, s_2, o were once randomly generated, $-5 < x_k < 5$ and $k = 1, 2, \ldots, 5$. The value of global minimum (global extreme) is -56.4176 and global extreme position is $x_1 = 4.0587$; $x_2 = -2.9964$; $x_3 = 2.7314$; $x_4 = -4.7486$ and $x_5 = -1.0560$. The function has 50 extremes randomly distributed in the space and one corresponding to the minimum of the hyperspace x^3.

4.4 Rastrigin Function

Rastrigin function is multimodal function that shows strong periodical character with the regular occurrence of extremes. It belongs to the separable test functions. The value of global minimum (global extreme) of this test function is 0, where $x_i = 0$ for $i = 1, 2, 3, \ldots, 5$; $x_i \in \langle -500; 500 \rangle$.

$$f(x) = 10n + \sum_{i=1}^{n} \left(x_i^2 - 10 \cos(2\Pi x_i) \right) . \tag{14}$$

success rate [%] SR -50	SUS	**sel3 proportional**	Tournament	sel3 order-based
high-div-local	2 %	**41 % p-sel = 1 p-rand = 0.5 and p-sel = 0.05 p-rand = 0.55 and p-sel = 0.15 p-rand = 1**	0 %	36 % p-sel = 1 p-rand = 0.45 and p-sel = 0.9 p-rand = 0.65 and p-sel = 1 p-rand = 0.8

Table 6. Fnc1 function – SR

mean best fitness MBF	SUS	**sel3 proportional**	Tournament	sel3 order-based
high-div-local	−25.256	**−37.568 p-sel = 0.05 p-rand = 0.55**	−16.243	−36.636 p-sel = 0.15 p-rand = 0.45

Table 7. Fnc1 function – MBF

success rate [%] SR 0,1	SUS	sel3 proportional	Tournament	**sel3 order-based**
mid-div-local	100 %	100 % p-sel= 1 p-rand = 0 and several others	99 %	**100 % p-sel = 1 p-rand = 0 and many others**
low-div-local	100 %	100 % p-sel = 1 p-rand = 0 and several others	100 %	**100 % p-sel = 1 p-rand** = 0 **and many others**

Table 8. Rastrigin function – SR

4.5 Sgu2 Function

Test function Sgu2 has a strong multimodal character and belongs to non-separable functions. The location and the value of global extreme is unknown for $x_i \in \langle -500; 500 \rangle$, $i = 1, 2, 3, \ldots, 5$. So far the best value achieved by different experiments was -46.2655.

$$f(x) = \sum_{i=1}^{n-1} - \left| \ln \left(\left| \arctan(x_{i+1}) - \arccos(x_i) \right| - \Pi \left(\sin(x_{i+1}) - \cos(x_i) \right) \right) \right| . \quad (15)$$

mean best fitness MBF	SUS	sel3 proportional	Tournament	**sel3 order-based**
mid-div-local	0.0005	0.0033 p-sel = 1 p-rand = 0	0.013	**0.0003** **p-sel = 1** **p-rand = 0**

Table 9. Rastrigin function – MBF

success rate [%] SR -25	SUS	**sel3 proportional**	Tournament	sel3 order-based
mid-div-local	61 %	**85 %** **p-sel = 0.7** **p-rand = 0**	73 %	78 % p-sel = 0.85 p-rand = 0.05

Table 10. Sgu2 function – SR

mean best fitness MBF	SUS	**sel3 proportional**	Tournament	sel3 order-based
mid-div-local	−27.209	**−28.189** **p-sel = 0.8** **p-rand = 0**	−27.181	−28.163 p-sel = 0.75 p-rand = 0

Table 11. Sgu2 function – MBF

success rate [%] SR -3650	SUS	sel3 proportional	Tournament	**sel3 order-based**
high-div-global	28 %	49 % p-sel = 0.8 p-rand = 0	15 %	48 % p-sel = 0.85 p-rand = 0
high-div-local	29 %	64 % p-sel = 0.65 p-rand = 0 and p-sel = 0.9 p-rand = 0.3	61 %	**67 %** **p-sel** = 0.65 **p-rand** = 015
mid-div-global	30 %	53 % p-sel = 0.6 p-rand = 0.1	46 %	54 % p-sel = 0.6 p-rand = 0.1
mid-div-local	25 %	61 % p-sel = 0.25 p-rand = 0.2 and p-sel = 0.55 p-rand = 0.7	45 %	61 % p-sel = 0.25 p-rand = 0
low-div-global	13 %	40 % p-sel = 0.6 p-rand = 0.65	18 %	37 % p-sel = 0.45 p-rand = 0.7
low-div-local	21 %	51 % p-sel = 0.15 p-rand = 0.85	28 %	46 % p-sel = 1 p-rand = 0.7

Table 12. Eggholder function – SR

mean best fitness MBF	SUS	sel3 proportional	Tournament	**sel3 order-based**
high-div-global	−3 314.42	−3 492.89 p-sel = 0.85 p-rand = 0	−3 421.74	−3 501.17 p-sel = 0.85 p-rand = 0
high-div-local	−3 274.18	−3 515.68 p-sel = 0.65 p-rand = 0.1	−3 502.99	**−3 538.92** **p-sel = 0.65** **p-rand = 0.15**
mid-div-global	−3 296.37	−3 513.28 p-sel = 0.6 p-rand = 0.1	−3 456.52	−3 509.35 p-sel = 0.6 p-rand = 0.35
mid-div-local	−3 251.13	−3 509.95 p-sel = 0.25 p-rand = 0.2	−3 392.8	−3 488.4 p-sel = 0.4 p-rand = 0.8
low-div-global	−3 129.94	−3 399.69 p-sel = 0.15 p-rand = 1	−3 185.14	−3 402.85 p-sel = 0.2 p-rand = 0.65
low-div-local	−3 137.88	−3 417.39 p-sel = 0.15 p-rand = 0.65	−3 229.15	−3 389.04 p-sel = 1 p-rand = 0.7

Table 13. Eggholder function – MBF

success rate [%] SR 0,1	SUS	**sel3 proportional**	Tournament	**sel3 order-based**
high-div-global	97 %	97 % p-sel = 1 p-rand = 0	0 %	99 % p-sel = 1 p-rand = 0
high-div-local	100 %	100 % p-sel = 1 p-rand = 0 and many others	100 %	100 % p-sel = 1 p-rand = 0 and many others
mid-div-global	100 %	100 % p-sel = 1 p-rand = 0 and p-sel = 1 p-rand = 0.05	32 %	100 % p-sel = 1 p-rand = 0 and several others
mid-div-local	100 %	100 % p-sel = 1 p-rand = 0 and many others	100 %	100 % p-sel = 1 p-rand = 0 and many others
low-div-global	96 %	97 % p-sel = 1 p-rand = 0.1	95 %	99 % p-sel = 0.55 p-rand = 0.05
low-div-local	100 %	**100 %** **all parameters** **p-sel, p-rand**	100 %	**100 %** **all parameters** **p-sel, p-rand**

Table 14. Quadratic function – SR

5 CONCLUSION

In this paper we introduced new selection method called 3-selection which enables higher scalability, covers both the proportional and order-based methods, and in addition, it has a randomness parameter. The examples show that for different settings of variation operators, which directly influence the diversity and the degree of searching, the meaning of selection method is changing.

mean best fitness MBF	SUS	sel3 proportional	Tournament	**sel3 order-based**
high-div-global	0.0269	0.0272 p-sel = 1 p-rand = 0	19.867	0.0195 p-sel = 1 p-rand = 0
high-div-local	0.0000412	0.000283 p-sel = 1 p-rand = 0	0.0064	0.00000369 p-sel = 1 p-rand = 0
mid-div-global	0.0092	0.0126 p-sel = 1 p-rand = 0	0.2481	0.0097 p-sel = 1 p-rand = 0
mid-div-local	0.0000126	0.0001259 p-sel = 1 p-rand = 0.15	0.0000653	**0.00000147** **p-sel = 0.95** **p-rand = 0**
low-div-global	0.0309745	0.0335 p-sel = 1 p-rand = 0.1	0.0375027	0.0284 p-sel = 0.75 p-rand = 0
low-div-local	0.0000123	0.0000512 p-sel = 1 p-rand = 0.1	0.00000746	0.00000448 p-sel = 0.95 p-rand = 0.05

Table 15. Quadratic function – MBF

success rate [%] SR -50	SUS	**sel3 proportional**	Tournament	sel3 order-based
high-div-global	12 %	31 % p-sel = 0.6 p-rand = 0.3	0 %	23 % p-sel = 1 p-rand = 0.15
high-div-local	2 %	**41 %** **p-sel = 1** **p-rand = 0.5** **and** **p-sel = 0.05** **p-rand = 0.55** **and** **p-sel = 0.15** **p-rand = 1**	0 %	36 % p-sel = 1 p-rand = 0.45 and p-sel = 0.9 p-rand = 0.65 and p-sel = 1 p-rand = 0.8
mid-div-global	12 %	27 % p-sel = 0.15 p-rand = 0.6	0 %	32 % p-sel = 1 p-rand = 0.3
mid-div-local	5 %	35 % p-sel = 0 p-rand = 0.4	0 %	38 % p-sel = 0 p-rand = 0.35
low-div-global	4 %	21 % p-sel = 0 p-rand = 0.2	0 %	20 % p-sel = 0 p-rand = 0.15
low-div-local	2 %	20 % p-sel = 0 p-rand = 0.65	0 %	20 % p-sel = 0 p-rand = 0.65

Table 16. Fnc1 function – SR

Compared with the most used fitness-proportionate selection method the SUS and the most used order-based selection method – the tournament selection, it was also shown that the proposed 3-selection method is able to provide better results because this method enables more precise setting of GA, and consequently, more precise results are obtained. Experiments used binary tournament selection with

mean best fitness MBF	SUS	**sel3 proportional**	Tournament	sel3 order-based
high-div-global	−28.972	−34.569 p-sel = 0.9 p-rand = 0.75	−17.274	−33.686 p-sel = 0.95 p-rand = 0.3
high-div-local	−25.256	**−37.568 p-sel = 0.05 p-rand = 0.55**	−16.243	−36.636 p-sel = 0.15 p-rand = 0.45
mid-div-global	−17.323	−34.621 p-sel = 0.05 p-rand = 0.5	−18.479	−34.4 p-sel = 0 p-rand = 0.6
mid-div-local	−15.175	−36.171 p-sel = 0 p-rand = 0.4	−18.121	−36.191 p-sel = 0 p-rand = 0.35
low-div-global	−11.042	−31.647 p-sel = 0 p-rand = 0.2	−17.926	−29.317 p-sel = 0 p-rand = 0.25
low-div-local	−9.029	−28.439 p-sel = 0 p-rand = 0.2	−9.615	−27.72 p-sel = 0 p-rand = 0.45

Table 17. Fnc1 function – MBF

success rate [%] SR 0,1	SUS	sel3 proportional	Tournament	**sel3 order-based**
high-div-global	0 %	0 %	0 %	0 %
high-div-local	100 %	100 % p-sel = 1 p-rand = 0	1 %	100 % p-sel = 1 p-rand = 0 and several others
mid-div-global	1 %	1 % p-sel = 1 p-rand = 0.05	0 %	3 % p-sel = 1 p-rand = 0
mid-div-local	100 %	100 % p-sel = 1 p-rand = 0 and several others	99 %	**100 % p-sel = 1 p-rand = 0 and many others**
low-div-global	0 %	1 % p-sel = 1 p-rand = 0.15 and several others	0 %	1 % p-sel = 0.45 p-rand = 0 and several others
low-div-local	100 %	100 % p-sel = 1 p-rand = 0 and several others	100 %	**100 % p-sel = 1 p-rand = 0 and many others**

Table 18. Rastrigin function – SR

arity = 2 and selection probability (for $k = 2$):

$$p(f_i) = \frac{(i_{\min}(f_i) + z(f_i))^k - (i_{\min}(f_i) + z(f_i) - 1)^k}{n^k} \tag{16}$$

mean best fitness MBF	SUS	sel3 proportional	Tournament	**sel3 order-based**
high-div-global	2.21	2.137 p-sel = 1 p-rand = 0	49.616	2.41 p-sel = 1 p-rand = 0
high-div-local	0.00107	0.018 p-sel = 1 p-rand = 0	3.864	0.0009 p-sel = 1 p-rand = 0
mid-div-global	1.488	1.403 p-sel = 1 p-rand = 0	6.971	1.385 p-sel = 1 p-rand = 0
mid-div-local	0.0005	0.0033 p-sel = 1 p-rand = 0	0.013	**0.0003** **p-sel = 1** **p-rand = 0**
low-div-global	2.582	2.6504 p-sel = 1 p-rand = 0.15	2.868	2.4198 p-sel = 0.9 p-rand = 0
low-div-local	0.0013	0.0027 p-sel = 1 p-rand = 0.2	0.0015	0.0008 p-sel = 1 p-rand = 0

Table 19. Rastrigin function – MBF

success rate [%] SR -25	SUS	**sel3 proportional**	Tournament	sel3 order-based
high-div-global	35 %	52 % p-sel = 0.85 p-rand = 0	3 %	49 % p-sel = 0.95 p-rand = 0
high-div-local	66 %	76 % p-sel = 0.8 p-rand = 0	16 %	79 % p-sel = 0.95 p-rand = 0
mid-div-global	47 %	57 % p-sel = 0.9 p-rand = 0	33 %	57 % p-sel = 0.8 p-rand = 0 and p-sel = 0.8 p-rand = 0.1
mid-div-local	61 %	**85 %** **p-sel = 0.7** **p-rand = 0**	73 %	78 % p-sel = 0.85 p-rand = 0.05
low-div-global	28 %	38 % p-sel = 0.85 p-rand = 0.1	20 %	33 % p-sel = 0.55 p-rand = 0.3
low-div-local	52 %	66 % p-sel = 0.95 p-rand = 0.8 and p-sel = 0.5 p-rand = 0.7	43 %	64 % p-sel = 0.3 p-rand = 0.25

Table 20. Sgu2 function – SR

where

- $p(f_i)$ is selection probability of the i^{th} individual (fitness),
- $z\left(f_i\right)$ is number of repetitions of the i^{th} individual (fitness) in the population,
- $i_{\min}(f_i)$ is number of individuals from which the i^{th} individual has better fitness (for minimization tasks)
- k is arity of the tournament selection,

mean best fitness MBF	SUS	**sel3 proportional**	Tournament	sel3 order-based
high-div-global	−23.801	−25.212 p-sel = 0.85 p-rand = 0	−20.472	−24.949 p-sel = 0.9 p-rand = 0
high-div-local	−26.282	−27.837 p-sel = 0.9 p-rand = 0	−22.263	−27.952 p-sel = 0.95 p-rand = 0
mid-div-global	−25.141	−25.555 p-sel = 0.9 p-rand = 0	−24.234	−25.871 p-sel = 0.8 p-rand = 0
mid-div-local	−27.209	**−28.189** **p-sel = 0.8** **p-rand = 0**	−27.181	−28.163 p-sel = 0.75 p-rand = 0
low-div-global	−23.005	−24.0403 p-sel = 0.85 p-rand = 0.1	−22.891	−23.893 p-sel = 0.7 p-rand = 0.6
low-div-local	−25.49	−26.5197 p-sel = 0.5 p-rand = 0.7	−24.877	−26.3822 p-sel = 0.3 p-rand = 0.25

Table 21. Sgu2 function – MBF

- n is number of individuals in the population.

The selection probability was calculated for used SUS:

$$p(X_i) = \frac{(f_{n_i} \times z(f_i))}{\sum f_{n_i}}; \quad f_{n_i} \geq 0 \tag{17}$$

where

- f_{n_i} is normalized fitness value of the i^{th} individual (minimization tasks),
- $p(X_i)$ is selection probability of the i^{th} individual,
- $z(f_i)$ is number of repetitions of the i^{th} individual (fitness) in the population.

It could be argued that the proposed 3-selection method makes the process of designing GA more difficult. By providing more degrees of freedom the process of designing GA is more difficult. However, the extra properties of selection predetermine this method to be used in some sort of adaptive algorithms where such disadvantages become useful. But the ability of continuously changing selection pressure, influence of randomness, or optionally changing behavior of selection by switching between proportional and order-based selection are definitely great merits of the 3-selection method.

Acknowledgement

This work has been supported by the Slovak Scientific Grant Agency VEGA under the grant No. 2/0155/19.

REFERENCES

[1] Vavak, F.—Fogarty, T. C.: Comparison of Steady State and Generational Genetic Algorithms for Use in Nonstationary Environments. Proceedings of IEEE International Conference on Evolutionary Computation, Nagoya University, 1996, pp. 192–195, doi: 10.1109/ICEC.1996.542359.

[2] Holland, J. H.: Adaptation in Natural and Artificial Systems. University of Michigan Press, 1975. ISBN 978-0-472-08460-9.

[3] Goldberg, D. E.—Deb, K.: A Comparative Analysis of Selection Schemes Used in Genetic Algorithms. Foundations of Genetic Algorithms, Vol. 1, 1991, pp. 69–93, doi: 10.1016/B978-0-08-050684-5.50008-2.

[4] Eiben, A. E.—Smith, J. E.: Introduction to Evolutionary Computing. Springer-Verlag, Berlin, Heidelberg, Natural Computing Series, 2003, doi: 10.1007/978-3-662-05094-1.

[5] Grefenstette, J. J.—Baker, J. E.: How Genetic Algorithms Work: A Critical Look at Implicit Parallelism. In: Schaffer, J. D. (Ed.): Proceedings of the Third International Conference on Genetic Algorithms. Morgan Kaufmann Publishers, San Mateo, CA, 1989, pp. 20–27.

[6] Sivaraj, R.—Ravichandran, T.: A Review of Selection Methods in Genetic Algorithm. International Journal of Engineering Science and Technology (IJEST), Vol. 3, 2011, No. 5, pp. 3792–3797.

[7] Shodhganga, `http://shodhganga.inflibnet.ac.in/bitstream/10603/32680/16/16/_chapter/%206.pdf`.

[8] Blickle, T.—Thiele, L.: A Comparison of Selection Schemes Used in Genetic Algorithms. Technical Report No. 11, Swiss Federal Institute of Technology (ETH), Computer Engineering and Communications Networks Lab (TIK), Zurich, 1995.

[9] De Jong, K. A.: An Analysis of the Behavior of a Class of Genetic Adaptive Systems. Doctoral Dissertation, Dissertation Abstracts International, Vol. 36, 1975, No. 10, 5140B, University of Michigan.

[10] Baker, J.: Reducing Bias and Inefficiency in the Selection Algorithm. Proceedings of the Second International Conference on Genetic Algorithms and Their Application, Hillsdale, New Jersey, 1987, pp. 14–21.

[11] Khachaturyan, A.—Semenovskaya, S.—Vainshtein, B.: Statistical-Thermodynamic Approach to Determination of Structure Amplitude Phases. Soviet Physics, Crystallography, Vol. 24, 1979, No. 5, pp. 519–524.

[12] Goldberg, D. E.: A Note on Boltzmann Tournament Selection for Genetic Algorithms and Population-Oriented Simulated Annealing. Complex Systems, Vol. 4, 1990, No. 4, pp. 445–460.

[13] Matsui, K.: New Selection Method to Improve the Population Diversity in Genetic Algorithms. 1999 IEEE Proceedings of the International Conference on Systems, Man, and Cybernetics (SMC '99), Vol. 1, 1999, pp. 265–630, doi: 10.1109/ICSMC.1999.814164.

[14] Baker, J. E.: Adaptive Selection Methods for Genetic Algorithms. In: Grefenstette, J. J. (Ed.): Proceedings of the International Conference on Genetic Algorithms. Hillsdale, 1985, pp. 101–111.

[15] Whitley, D.: The GENITOR Algorithm and Selection Pressure: Why Rank Based Allocation of Reproductive Trials Is Best. In: Schaffer, J. D. (Ed.): Proceedings of the Third International Conference on Genetic Algorithms. Morgan Kaufmann Publishers, San Mateo, CA, 1989, pp. 116–121.

[16] Bandyopadhyay, S.—Murthy, C. A.—Pal, S. K.: Pattern Classification Using Genetic Algorithms. Pattern Recognition Letters, Vol. 16, 1995, No. 8, pp. 801–808, doi: 10.1016/0167-8655(95)00052-I.

[17] Fernandes, C.—Tavares, R.—Munteanu, C.—Rosa, A.: Using Assortative Mating in Genetic Algorithms for Vector Quantization Problems. Proceedings of the 2001 ACM Symposium on Applied Computing (SAC 2001), ACM, 2001, pp. 361–365, doi: 10.1145/372202.372367.

[18] Huang, C.-F.: An Analysis of Mate Selection in Genetic Algorithms. Technical Report CSCS-2001-002, Center for the Study of Complex Systems, University of Michigan, 2001.

[19] Ochoa, G.—Mädler-Kron, C.—Rodriguez, R.—Jaffe, K.: Assortative Mating in Genetic Algorithms for Dynamic Problems. In: Rothlauf, F., Branke, J., Cagnoni, S. et al. (Eds.): Applications of Evolutionary Computing (EvoWorkshops 2005). Springer, Berlin, Heidelberg, Lecture Notes in Computer Science, Vol. 3449, 2005, pp. 617–622, doi: 10.1007/978-3-540-32003-6_65.

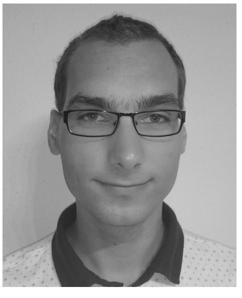
[20] Eshelman, L. J.—Schaffer, J. D.: Preventing Premature Convergence in Genetic Algorithms by Preventing Incest. Proceedings of the 4$^{\text{th}}$ International Conference on Genetic Algorithms (ICGA '91), Morgan Kaufmann, San Francisco, CA, 1991, pp. 115–122.

[21] Hutter, M.: Fitness Uniform Selection to Preserve Genetic Diversity. Proceedings of the 2002 IEEE Congress on Evolutionary Computation (CEC '02), 2002, pp. 783–788, doi: 10.1109/CEC.2002.1007025.

[22] Chen, Y.—Hu, J.—Hirasawa, K.—Yu, S.: Performance Tuning of Genetic Algorithms with Reserve Selection. Proceedings of the 2007 IEEE Congress on Evolutionary Computation (CEC '07), 2007, pp. 2202–2209, doi: 10.1109/CEC.2007.4424745.

[23] Bäck, T.: Selective Pressure in Evolutionary Algorithms: A Characterization of Selection Mechanisms. Proceedings of the First IEEE Conference on Evolutionary Computation, IEEE World Congress on Computational Intelligence (ICEC '94), Vol. 1, 1994, pp. 57–62, doi: 10.1109/ICEC.1994.350042.

Jana Lovíšková works as a researcher at the Institute of Informatics of the Slovak Academy of Sciences in the Department of Modelling and Control of Discrete Processes. She received her Master's (2001) and Ph.D. (2010) degrees in the field of industrial management from the Faculty of Materials Science and Technology of the Slovak Technical University in Bratislava. Her research interests include optimization, search and scheduling problems solution by means of evolutionary computing, mainly genetic algorithms.



Daniel Pernecký received his Master's degree in cybernetics at Slovak University of Technology in Bratislava, Faculty of Electrical Engineering and Information Technology in 2014. His main resarch area is an optimization by metaheuristic methods algorithms.

Computing and Informatics, Vol. 38, 2019, 1444–1478, doi: 10.31577/cai_2019_5_1444

CHAOTIC ELECTION ALGORITHM

Hojjat Emami

Computer Engineering Department, University of Bonab
Bonab, East Azerbaijan, Iran
e-mail: `emami@ubonab.ac.ir`

Abstract. A novel Chaotic Election Algorithm (CEA) is presented for numerical function optimization. CEA is a powerful enhancement of election algorithm. The election algorithm is a socio-politically inspired strategy that mimics the behavior of candidates and voters in presidential election process. In election algorithm, individuals are organized as electoral parties. Advertising campaign forms the basis of the algorithm in which individuals interact or compete with one other using three operators: positive advertisement, negative advertisement and coalition. Advertising campaign hopefully causes the individuals converge to the global optimum point in solution space. However, election algorithm suffers from a fundamental challenge: it gets stuck at local optima due to the inability of advertising campaign in searching solution space. CEA enhances the election algorithm through modifying party formation step, introducing chaotic positive advertisement and migration operator. By chaotic positive advertisement, CEA exploits the entire solution space, what increases the probability of obtaining global optimum point. By migration, CEA increases the diversity of the population and prevents early convergence of the individuals. The proposed CEA algorithm is tested on 28 well-known standard boundary-constrained test functions, and the results are verified by a comparative study with several well-known meta-heuristics. The results demonstrate that CEA is able to provide significant improvement over canonical election algorithm and other comparable algorithms.

Keywords: Optimization, meta-heuristic, election algorithm, Chaotic Election Algorithm (CEA)

Mathematics Subject Classification 2010: 68T01, 68T20, 68T05, 68W25

1 INTRODUCTION

Optimization is the process of finding the best solution from among the set of all feasible solutions subject to a given set of constraints. An optimization problem can be represented as a minimization (maximization) model with the goal to obtain a point x^* from a solution space $S \in R^n$, where objective function $f : S \rightarrow R$ is minimized, i.e. $f(x^*) \leq f(x)$ for all $x \in S$. In recent years, several meta-heuristics have been presented to solve optimization problems. A bibliography of recently proposed meta-heuristics is given in [1], and several surveys are given in [2, 3, 4, 5, 6, 7]. Generally speaking, meta-heuristics can be classified into three main categories [53]: evolutionary, swarm intelligence and physics-based algorithms.

Evolutionary meta-heuristics are mainly inspired by the concepts of natural biological evolution, in which the fittest individuals can survive and the weak must die [8]. In natural evolution survival is achieved through reproduction. Evolutionary algorithms begin their optimization process with a randomly generated population of individuals, where any individual is a candidate solution for the given problem. For each generation, individuals compete with each other to reproduce offspring. The best-fit individuals have the best chance to reproduce. Offspring are generated by the combination and mutation of the individuals in the previous generation. The offspring iteratively update over the course of generations until an optimal solution is reached. Some of the well-known evolutionary algorithms are Genetic Algorithm (GA) [9], Differential Evolution (DE) [35], Biogeography-Based Optimizer (BBO) [42] and Backtracking Search Optimization Algorithm (BSA) [10].

The second main branch of meta-heuristics is swarm intelligence algorithms. These algorithms are inspired by natural or non-natural phenomena and mostly mimic the social behavior of swarms and social organisms [53, 8]. For example, Artificial Bee Colony (ABC) [11] is a nature inspired algorithm, which models intelligent behavior of honey bees in nature. Another example is election algorithm [12], a non-natural inspired algorithm, which simulates candidates' behavior in a presidential election process. Swarm intelligence based algorithms are multi-agent models. These algorithms model the intelligent behaviors of agents and their local interaction with the environment and neighboring agents to explore solution space and reach global optima. Some of the well-known swarm intelligence algorithms include: Particle Swarm Optimization (PSO) [13], Ant Colony Optimization (ACO) [37], ABC [11], Election algorithm [12], Firefly Algorithm (FA) [44], Grey Wolf Optimizer (GWO) [53] and Salp Swarm Algorithm (SSA) [55].

The third class of meta-heuristics is physics-based methods, which almost mimic the physical processes of nature. For example, Big-Bang Big-Crunch (BB-BC) [28] is inspired by the evolution of universe; and Gravitational Search Algorithm (GSA) [43] is developed based on gravity law. Some other well-known algorithms that fall into the category of physics-based meta-heuristics include: Intelligent Water Drops (IWD) [15], Charged System Search (CSS) [16], Black Hole (BH) [17] and Magnetic Optimization Algorithm (MOA) [18]. For a survey of physics-based algorithms see [19].

Meta-heuristics are widely used in various scientific and engineering applications because they have shown good performance in solving large-scale, complex non-linear and non-differentiable problems. The applications range from data mining [20], image processing [21] and social network analysis [22] in computer science domain, at one end of the spectrum, to air traffic control [23], airfoil design [55], optical buffer design [53] in industrial field, at the other side of spectrum. However, according to the famous "No Free Lunch" theory [24], there is no meta-heuristic best suited for solving all optimization problems. A particular meta-heuristic may show promising results on a set of problems, but the same algorithm may show poor results on a different set of problems. On the other hand, meta-heuristics achieved encouraging results on optimization problems but their performance far from the ideal. According to this issue and the NFL theory, it is obvious that there is still a room for introducing new meta-heuristics or improving existing meta-heuristics.

As an element of research in this field, this paper presents a new Chaotic Election Algorithm, denoted as CEA. The CEA enhances the canonical election algorithm threefold:

1. increasing the speed of party formation step employing random initialization method,
2. introducing migration operator to enhance the diversity of population and preventing early convergence of the algorithm, and
3. introducing chaotic positive advertisement operator to searching efficiently the entire solution space.

The CEA algorithm is tested on 28 test problems and compared with several well-known meta-heuristics. The experimental results show that the proposed algorithm outperforms counterpart meta-heuristics for several benchmark test functions.

The rest of the paper is organized as follows. Section 2 presents related work, with the focus on chaotic swarm optimization algorithms. Section 3 presents the canonical election algorithm. Section 4 outlines the proposed Chaotic Election Algorithm (CEA). In Section 5, the proposed algorithm is tested on numerical optimization benchmark problems and the simulation results are compared with several well-known algorithms. Finally, Section 6 presents a conclusion of this work and suggests some directions for future work.

2 RELATED WORK

Most of the meta-heuristic algorithms suffer from stagnation in local optima and low convergence rate. With the development of the nonlinear dynamics, chaos theory has been widely used in various applications [29]. One of the major applications is the introduction of chaos concept into the optimization meta-heuristics. Chaos mechanism is one of the best methods to improve the performance of evolutionary algorithms in terms of both local optima avoidance and convergence speed [32]. Due to the ergodicity and randomness nature, chaos has several advantages that include self-organization, evolution, easy implementation and high ability to avoid being trapped in local optima [34, 41]. Due to these properties, simultaneous use of chaos and optimization algorithms improves the performance of algorithms. Up to now, the chaos theory has been successfully combined with several meta-heuristic optimization methods [41]. Table 1 lists some familiar meta-heuristics and their improved ones by incorporated chaos. It is important to notice that Table 1 is not aiming to summarize a comprehensive survey of such chaotic combination, but to show that utilizing chaotic mechanisms indeed empowers the algorithm to possess better performance. This issue highlights that there is an interesting room to combine other meta-heuristics with chaotic mechanism to improve their performance.

Algorithm	Reference	
	Canonical Version	Chaotic Version
Differential Evolution	[35]	[36]
Ant Colony Optimization	[37]	[38]
Artificial Bee Colony Algorithm	[39]	[31]
Imperialist Competitive Algorithm	[40]	[41]
Biogeography-Based Optimization	[42]	[32]
Gravitational Search Algorithm	[43]	[30]
Bat Swarm Optimization	[44]	[45]
Cuckoo Search Algorithm	[46]	[47]
Firefly Algorithm	[48]	[49]
Particle Swarm Optimization	[50]	[51]
Krill Herd Algorithm	[52]	[29]
Grey Wolf Optimizer	[53]	[34, 54]
Salp Swarm Algorithm	[55]	[56]

Table 1. Some meta-heuristics and their corresponding chaotic meta-heuristics

Jia et al. [36] proposed DECLS algorithm to enhance the search ability of Differential Evolution (DE). DECLS explores a huge search space in the early run phases to avoid premature convergence, and exploiting a small region in the later run phases to refine the final solutions. Cai et al. [38] proposed Chaotic Ant Swarm Optimization (CASO) algorithm for solving the economic dispatch problems of thermal generators in power systems. CASO combines the chaotic and swarm-based search capability of ants in searching the global optimum solution.

Alatas [31] proposed Chaotic ABC (CABC) algorithm that adopts chaotic maps for parameter adaptation to prevent the ABC to get stuck on local optima and to improve its convergence speed. This is done by using of chaotic number generators each time a random number is needed by the canonical ABC algorithm.

Talatahari et al. [41] proposed a Chaotic Imperialist Competitive Algorithm (CICA). They used different chaotic maps to improve the assimilation phase of the algorithm. The results on four benchmark problems show the benefits of using chaotic maps in assimilation phase. Saremi et al. [32] investigated the effectiveness of ten different chaotic maps in solving the entrapment in local optima and slow convergence speed problems of the BBO algorithm. They used chaotic maps to define selection, emigration, and mutation probabilities. The experiments show that the chaotic maps are able to improve the performance of BBO.

Gao et al. [30] proposed Chaotic Gravitation Search Algorithms (CGSA) to alleviate the slow convergence and local optima trapping problems of GSA algorithm. The big problem in the canonical Bat Swarm Optimization (BSO) is the premature convergence into local optima. To alleviate this issue, Rezaee [45] presented the CBSO algorithm, which is a chaotic-based bat swarm optimization algorithm. In CBSO, the loudness is updated via multiplying a linearly decreasing function by chaotic map functions.

Wang et al. [47] proposed Chaotic Cuckoo Search (CCS) that embeds chaotic mechanisms into Cuckoo Search (CS) algorithm. In CCS, twelve chaotic maps are applied to tune the step size of the cuckoos used in the original CS algorithm. The experiments on optimization benchmark problems show that the performance of CCS is much better than canonical CS algorithm.

Gandomi et al. [49] proposed Chaotic Firefly Algorithm (CFA) algorithm that incorporated chaos into FA so as to increase its global search mobility. They used twelve different chaotic maps to tune the attractive movement of the fireflies in the algorithm. The experiments show that CFA outperforms the canonical FA.

Alatas et al. [51] proposed twelve different Chaos Embedded Particle Swarm Optimization Algorithms (CEPSOAs) that use chaotic maps for parameter adaptation. CEPSOAs use chaotic number generators each time a random number is needed by the canonical PSO algorithm. The results on benchmark problems show that CEPSOAs increased the solution quality and improved the global searching capability by escaping the local optimum points.

Yaghoobi and Mojallali proposed an Improved Chaotic Krill Herd (ICKH) algorithm used for PID controller design [57]. The main idea of the ICKH is to combine chaos theory and Krill Herd (KH) algorithm to improve the search efficiency.

Yu et al. [34] incorporated chaotic local search mechanism to enhance the search dynamics of GWO algorithm and accelerating it convergence speed. They investigated twelve different kinds of chaotic maps to identify the influence of chaotic search capability on GWO. The results show that chaotic empowers GWO to achieve better performance in terms of solution quality and convergence speed. In another work, Kohli and Arora [54] proposed CGWO algorithm that uses different chaotic maps to regulate the key parameter "a" of GWO algorithm, with the aim of accelerating its convergence speed. The results show the superiority of CGWO when compared to GWO algorithm.

In order to boost the performance of the canonical SSA, Sayed et al. [56] proposed Chaotic Salp Swarm Algorithm (CSSA) that is a hybrid solution based on SSA algorithm and chaos theory. They evaluated ten chaotic maps and found that logistic chaotic map is the optimal map of the used ten maps. The simulation results on optimization benchmarks and feature selection problem reveal the superiority of CSSA algorithm when compared to canonical SSA and some other counterparts.

After this short review, and from the experimental studies presented in the above-mentioned literature, it is obvious that utilizing chaotic mechanisms indeed empowers the algorithm to get better results. This issue highlights that there is an interesting room to combine other meta-heuristics with chaotic mechanism to improve their performance.

3 ELECTION ALGORITHM

3.1 General Aspects

The election algorithm simulates the socio-political process of presidential election in real world [12]. It is a multi-agent algorithm, in which agents are called "persons". There are two types of persons: candidates and voters. Some of the best persons are selected to be the candidates and the remaining are the voters. Initially, all the voters are divided among the candidates based on their similarity in opinions and ideas. Candidates together with their voters form some political parties.

Once initial parties are formed, the candidates start their advertising campaign. Candidates to advertise themselves employ two kinds of advertisements: positive advertisement and negative advertisement. In positive advertisement, candidates convey their agendas and ideas to the voters and attempt to attract the voters towards themselves. In negative advertisement, candidates attempt to increase their own popularity and decrease the popularity of other candidates. Any candidate that is not able to succeed in negative advertisement and cannot increase his popularity will be eliminated. The candidates that have similar opinions can unite and form a new party which is a combination of these parties. This process is a simple model of coalition which is pursued by some candidates in real-world elections. The election algorithm iteratively applies positive advertisement, negative advertisement and coalition on population until termination conditions are satisfied. Once the algorithm stops, the candidate who attained the majority of votes will be announced as the winner. The winner candidate is equal to the best solution found for the given optimization problem.

3.2 Working Principle

Figure 1 shows the working principle of the election algorithm. The algorithm starts with an initial population. Each individual in the population is called a person. For a problem with x_1, x_2, $\dots$, $x_{N_{var}}$ variables, the initial population consists of N_{pop} persons. Each person P_i is an $1 \times N_{var}$ array of variables values and is defined as

$$P_i = [x_1, x_2, \dots, x_{N_{var}}]. \tag{1}$$

The eligibility of a person P_i is found by evaluation of the eligibility function $\mathbf{E}$ at the variables x_1, x_2, $\dots$, $x_{N_{var}}$ considering objective function of the problem.

The eligibility function is defined as follows:

$$E(P_i) = E(x_1, x_2, \ldots, x_{N_{var}}). \tag{2}$$

The persons are divided to several political parties. To fulfill this aim, from the total population, N_c of the most popular persons (the persons with best eligibility values) are selected to be candidates, and the remaining N_v persons will be the voters, each of which belongs to a candidate. The voters are divided among candidates based on their eligibility distance. Voter v_k is considered as a supporter of candidate c_i, if the following predicate holds.

$$P_i = \{v_k : |E_{v_k} - E_{c_i}| < |E_{v_k} - E_{c_j}| \quad \forall \;\; 1 \leq j \leq N_c\} \tag{3}$$

where P_i is the i^{th} party and N_c is the number of initial candidates. E_{c_i} and E_{v_k} present the eligibility of candidate c_i and voter v_k, respectively. In the party formation process, each voter is assigned to exactly one party. After dividing the voters among candidates and forming the initial parties, the candidates start advertising campaign. The advertising campaign consists of three main phases: positive advertisement, negative advertisement and coalition.

The positive advertisement is modeled by conveying some variables of the candidate to its voters inside a party. To do this task, in each party, N_s variables of the target candidate are randomly selected and replaced with the selected variables of the voters. N_s is computed as follows:

$$N_s = \lceil X_s \times S_c \rceil \tag{4}$$

where S_c is the number of candidate's variables and X_s is the selection rate. The selected variables N_s are weighted with a coefficient ω and then embedded in voters. The new value for the i^{th} variable of a voter after positive advertisement is given by:

$$x_{i_{new}} = \omega . x_{i_{old}}, \quad \text{where} \quad \omega = \frac{1}{|E_{c_i} - E_{v_k}| + 1}. \tag{5}$$

In negative advertisement, candidates try to attract voters of weak candidates toward themselves. A party is weak if its candidate to be the weakest compared to other parties' candidates. To model the negative advertisement, first, a number of voters from the weakest party are selected. Then, a race is taking place among powerful parties to possess these voters. To select the weakest voters from the weakest party, the eligibility distance between the voters, and the weakest candidate is computed, and then 5 % of the farthest voters are selected. The distances between selected voters and the powerful candidates are computed, and the voters are assigned to the closest candidates.

In coalition phase, several candidates join together and form a new party. Among the candidates that wish to collate, a candidate is picked up at random to be the leader candidate and the remaining are considered as the followers. In coalition, all of the follower candidates and their voters become the voters of the leader one.

Until termination conditions are not satisfied, the advertising campaign operators are iteratively applied to update the population. Finally, the update process stops and the candidate with the majority of votes is announced as the winner. The winner is equal to the best solution found for the optimization problem.

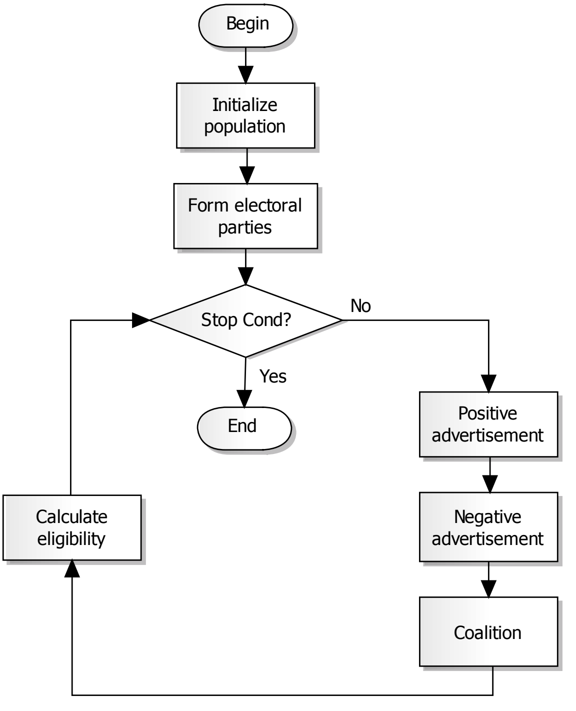


Figure 1. The working principle of the election algorithm

4 CHAOTIC ELECTION ALGORITHM

The advertising campaign is the core operator in the election algorithm, which causes the individuals converge to an optimal point in the search space. However, advertising campaign suffers from three challenges:

1. computing the Euclidean distance in the creation of initial parties and the negative advertisement steps that decrease the speed of the algorithm,
2. getting stuck at local optima,
3. inefficiency of positive advertisement phase.

In advertising campaign, after several iterations, diversity in the population may decrease. As a result, the candidates and their voters cannot explore the entire

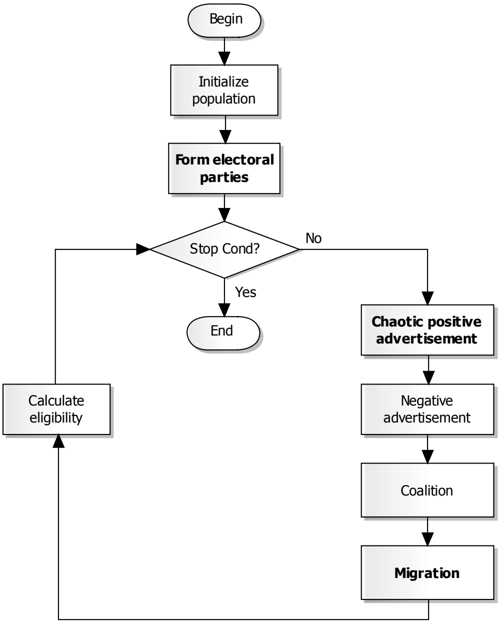


Figure 2. The working principle of the CEA algorithm

solution space and get stuck at local optima. To alleviate these issues, we proposed a Chaotic Election Algorithm, denoted as CEA. Figure 2 shows the flowchart of the CEA algorithm. The CEA enhances the election algorithm threefold:

1. increasing the speed of electoral party formation step utilizing random initialization method,
2. introducing migration operator, and
3. improving the positive advertisement using chaotic maps.

In the following, these enhancements are described.

4.1 Electoral Party Formation

As mentioned above, one drawback of the election algorithm is the computation of Euclidean distance for creating the initial electoral parties that decreases the speed of the algorithm. To alleviate this issue, we substitute the computation of Euclidean distance with a random initialization process. By this way, the voters are divided among candidates based on their eligibility, in which the initial number of voters of a candidate is proportionate to its eligibility. To identify the voters of a candidate c_i, first, its normalized eligibility is computed as

$$ne_{c_i} = \left| \frac{e_{c_i} - \max(I)}{\sum k \in N_c e_{c_k} - \max(I)} \right| \quad \text{where} \quad I = \{e_{c_j} | j \in N_c\} \tag{6}$$

where e_{c_i} is the eligibility of candidate c_i, ne_{c_i} is the normalized eligibility of candidate c_i, and N_c is the initial number of candidates. The initial number of voters of candidate c_i is computed as

$$N_{v_{c_i}} = \lceil ne_{c_i} \times N_v \rceil . \tag{7}$$

N_v is the number of all voters.

Then we randomly select $N_{v_{c_i}}$ of the voters and give them to candidate c_i. The voters along with their candidate c_i form an electoral party P_i in the solution space.

4.2 Migration

We introduced migration operator to help the election algorithm maintain diversity in the population and improve its optimization and search capability. Migration keeps the election algorithm away from converging too fast before exploring the entire solution space. The motivation to introducing the migration operator comes from the fact that in some real-world elections, some individuals can travel from other countries to the target country and vote to their favourite candidate. The travellers are referred as migrants, which can increase the popularity of some candidates. To model migration, some new voters are randomly generated on different areas of the solution space. Here, the new generated voters referred as migrants. The number of migrants at every generation of the algorithm is given by:

$$M = \lceil \mu \times N_{pop} \rceil \tag{8}$$

where M is the number of new migrants, μ is the migration coefficient, and N_{pop} is the population size. In the implementations, the proper value for μ is determined empirically. The migration in every generation of the algorithm adds M new individuals to the population. This causes two issues:

1. excessive growth of the population and
2. increasing the computational time of the algorithm.

To alleviate these issues, we eliminate M of the weakest individuals from the population at every generation of the algorithm. To do this, first all of the individuals in the population are sorted based on their eligibility in ascending order and then M of the inferior individuals (the individuals with lowest eligibility) are removed.

4.3 Chaotic Positive Advertisement

In the election algorithm, positive advertisement is realized through transferring some randomly selected variables from a candidate to its voters. The information only transfers towards voters and the candidate remains without change. Two weaknesses may exist in this way. First, the information exchange (social learning) is one-directional, in which some variables of candidates convey towards voters. As a result, the candidates and their voters cannot explore the entire solution space and the convergence speed decreases. Second, the voters who are affected and their variables are all chosen randomly. As a result, voters with higher eligibility, which may guide the population towards global optimums are not utilized. To overcome these issues and improve the exploration and exploitation ability of canonical EA, we proposed a new chaotic positive advertisement. Chaos is a special kind of dynamic behavior of non-linear systems [41]. Due to the high ability to avoid being trapped in local optima and easy implementation, chaos has raised enormous interest in optimization theory [41, 57]. The application of chaotic maps instead of random variables in the positive advertisement phase is a powerful mechanism to increase diversity of the population and improve the CEA's performance in preventing premature convergence to local optima. Let $v_k(t)$ denote the position of voter k in the search space at iteration t, and $c_i(t)$ denote the position of candidate i at iteration t. The position of voter v_k at iteration $t+1$ is computed as

$$v_k\,(t+1) = v_k(t) + A + B \tag{9}$$

where t indicates the current iteration, A and B are coefficient vectors, which are calculated as

$$A = \omega \times r \times V_1, \tag{10}$$

$$B = \omega \times r \times \tan\,(\theta) \times V_2 \tag{11}$$

where r is the chaotic variable generated based on a chaotic map, V_1 is a vector where its starting point is the previous position of the voter v_k and its direction is toward the candidate position c_i, and V_2 is a unit vector which is perpendicular to V_1. It is important to notice that $V_1.V_2 = 0$. ω is the distance between voter v_k and candidate c_i, which is computed as

$$\omega = \left|c_i(t) - v_k(t)\right|. \tag{12}$$

By term A, the candidate c_i attracts voter v_k towards itself with no deviation (point l_1 in Figure 3). In order to increase the searching around the candidate c_i, some deviations are added to locate the final position of the voter v_k in its movement toward candidate c_i (point l_2 in Figure 3). By this way, different points around the candidate c_i are explored. $\theta \in U(-\lambda, +\lambda)$ is a random number with uniform distribution regenerated every iteration. λ adjusts the deviation of voter v_k from its original direction. In our implementation, $\lambda = \pi/4$ is used that resulted in good convergence of individuals to the global optimum.

Different chaotic maps can be used to generate chaotic variables. In our implementations, we used logistic map [41] to generate chaotic variable r. The reason to this choice is that CEA have shown better performance when logistic map have been used in compared to the other chaotic maps. The logistic map shows good chaotic properties, it displays better randomness than other maps, and it can navigate the algorithm to the points that have been distributed in search space as much as possible [30, 27].

Logistic map is defined as

$$r_{k+1} = ar_k(1 - r_k) \tag{13}$$

where r_k represents the k^{rmth} number in the chaotic sequence, and k means the index of the chaotic sequence. $r \in (0, 1)$ under the conditions that the initial $r_0 \in (0, 1)$ and that $r_0 \notin \{0.0, 0.25, 0.5, 0.75, 1\}$. In the experiments $a = 4$ is used. In the current study, 1-dimension, non-invertible logistic map is used to produce chaotic sequences.

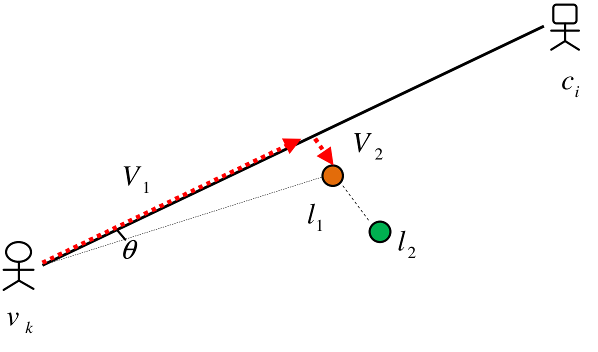


Figure 3. Attracting of voter v_k toward candidate c_i in the chaotic positive advertisement

Due to the non-repetition and ergodicity property of chaotic variables and non-repetition nature of chaos, the newly proposed chaotic positive advertisement carries out overall searches at higher speed than the standard positive advertisement, which is based on the random-based searches. The incorporation of the chaotic positive advertisement in the CEA has two advantages: (*i*) improving the information exchange between candidates and voters, and (*ii*) searching efficiently the entire solution space to find a global optimum point. Based on the simulation results presented in the next section, CEA is faster when compared with the canonical EA.

5 EXPERIMENTS

The proposed CEA algorithm is tested on 28 benchmark functions. The CEA algorithm is compared with several top-performing meta-heuristics in solving real-parameter optimization problems, including Covariance Matrix Adaptation Evolution Strategies (CMA-ES) [60], Self-adaptive Differential Evolution (SaDE) algorithm [61], adaptive Differential Evolution (JDE) algorithm [62], PSO2011 [50], Election algorithm Emami2015, Socio Evolution & Learning Optimization (SELO) algorithm [63], and Chaotic Salp Swarm Algorithm (CSSA) [56]. CMA-ES, SaDE, and JDE algorithms are the most successful optimization algorithms. In the competitions at different CEC conferences, these algorithm and their variants possess top positions when compared to other best performing algorithms. PSO2011 [50] is an advanced version of the standard PSO, which incorporates many improvements of PSO that have been identified by years of studies. SELO is a novel meta-heuristic inspired by the social learning behavior of humans organized as families in a societal setup. The reason behind the selection of SELO as a comparative algorithm is that it is a socio-inspired strategy (similar to CEA, which is a socio-politically inspired strategy), and it outperformed other socio-inspired algorithms. CSSA is a chaotic version of SSA algorithm and achieved encouraging results [56].

5.1 Benchmark Functions

Twenty eight well-known benchmark functions are used in the experiment. These are continuous, unbiased optimization problems and have different degrees of complexity and multi-modality. This set of problems has different kinds of properties such as unimodal, multimodal, separable and non-separable. These problems are single objective optimization problems taken from various sources including CEC2005 [58], CEC2013 [64], CEC2015 [59] and recently published papers. The benchmark functions can be classified into four groups:

Group I: F1-F10 are unimodal functions. These functions are used to assess the fast-converging performance of CEA and comparative algorithms.

Group II: F11-F20 are multimodal functions. These functions have many local optima points and are considered to evaluate the ability of algorithms to avoid local optima. Details about hybrid benchmarks are given in [11, 39].

Group III: F21-F24 are shifted and rotated multimodal functions whose base functions belong to Group II functions. These functions are enough complex and used to test the search capability of algorithms.

Group IV: F25-F28 are hybrid multimodal functions whose base functions belong to Group I, II and III functions. This set of functions is more complex than other ones and used to test the performance of algorithms in finding the global optimum of problems consisting of different subcomponents with different properties. Details about hybrid benchmarks are given in [58, 59].

We test the benchmark functions in 30 and 50 dimensions to draw empirical conclusion on the performance of the algorithms. Tables 2, 3, 4 and 5 list the characteristics of benchmark functions used in the tests. These functions are introduced in evolutionary computation share tasks and utilized by many researchers [11, 63, 10, 53].

We have chosen these benchmarks to be able to fairly compare our results to those of the counterpart algorithms.

5.2 Parameter Setting

The initial population sizes of all algorithms were 100 and the maximum number of function evaluations was 100 000. The other specific parameters for the algorithms are given in Table 6, as provided by their authors. We used five predefined criteria to terminate the algorithms' searching process that include:

- if the algorithm failed to find a better solution than the existing solution during the last 100 000 function evaluations,
- if the number of function evaluations reaches 1 000 000,
- if the maximum number of iterations (2 000 000 iterations) was reached,
- if the value of the objective function is less than 10^{-16},
- if the fitness value reaches below a predefined maximum error, the function evaluation is terminated.

All algorithms were programmed in MATLAB R2017a on a Personal Computer Intel Pentium 4 with the 3 GHz and 2 GB RAM. The operating system of the computer is Windows 7.

Problem	Name	Type	Range	Minimum	Definition
f_1	Cigar	S	[0, 10]	0	$f_1(x) = x_1^2 + 10^6 \sum_{i=2}^{n} x_i^2$
f_2	Discus	S	[0, 10]	0	$f_2(x) = 10^6 x_1^2 + \sum_{i=2}^{n} x_i^2$
f_3	DixonPrice	N	[-10, 10]	0	$f_3(x) = (x_1 - 1)^2 + \sum_{i=2}^{n} i(2x_i^2 - x_{i-1})^2$
f_4	Powell	N	[-4, 5]	0	$f_4(x) = \sum_{i=1}^{n} x_i^2 + \left(\sum_{i=1}^{n} 0.5 i x_i\right)^2 + \left(\sum_{i=1}^{n} 0.5 i x_i\right)^4$
f_5	Rosenbrock	N	[-30, 30]	0	$F_5(x) = \sum_{i=1}^{n-1} \left[100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2\right]$
f_6	Schwefel_1_2	N	[-100, 100]	0	$f_6(x) = \sum_{i=1}^{n} \left(\sum_{j=1}^{i} x_j\right)^2$
f_7	Schwefel_2_22	N	[-10, 10]	0	$f_7(x) = \sum_{i=1}^{n} \lvert x_i \rvert + \prod_{i=1}^{n} \lvert x_i \rvert$
f_8	Sphere	S	[-100, 100]	0	$f_8(x) = \sum_{i=1}^{n} x_i^2$
f_9	Sumsquares	S	[-10, 10]	0	$f_9(x) = \sum_{i=1}^{n} i x_i^2$
f_{10}	Zakharov	N	[-5, 10]	0	$f_{10}(x) = \sum_{i=1}^{n} x_i^2 + \left(\sum_{i=1}^{n} 0.5 i x_i\right)^2 + \left(\sum_{i=1}^{n} 0.5 i x_i\right)^4$

Table 2. Unimodal benchmark problems. Range: limits of search space, N: non-separabple, S: separable.

Problem	Name	Type	Range	Minimum	Definition
f_{11}	Ackley	N	[-32, 32]	0	$f_{11}(x) = -20\exp\left(-0.02\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}\right) - \exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)\right) + 20 + e$
f_{12}	Alpine	N	[0, 10]	0	$f_{12} = \sum_{i=1}^{n} \lvert x_i sin(x_i) + 0.1x_i \rvert$
F_{13}	Griewank	N	[-600, 600]	0	$F_{13}(x) = \frac{1}{4000}\left(\sum_{i=1}^{n} x_i^2\right) - \left(\prod_{i=1}^{n}\cos\left(\frac{x_i}{\sqrt{i}}\right) + 1\right)$
f_{14}	Leavy	N	[-10, 10]	0	$f_{14}(x) = \sin^2(\pi w_1) + \sum_{i=1}^{n-1}(w_i - 1)^2\left[1 + 10\sin^2(\pi w_i + 1)\right] + (w_n - 1)^2\left[1 + \sin^2(2\pi w_n + 1)\right]$ $w_i = 1 + \frac{x_i - 1}{4}, \quad i = 1, 2, \ldots, n$
f_{15}	Penalized	N	[50, 50]	0	$f_{15}(y) = \frac{\pi}{n} \times \left\{10\sin^2(\pi y_1) + \sum_{i=1}^{n-1}(y_i - 1)^2\left[1 + 10\sin^2(\pi y_{i+1})\right] + (y_n - 1)^2\right\} +$ $\sum_{i=1}^{n} u(x_i, a, k, m)$ $(y_i = 1 + \frac{1}{4}(x_i + 1), \quad u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m & \text{if } x_i > a \\ 0 & \text{if } -a \le x_i \le a \\ k(-x_i - a)^m & \text{if } x_i < -a \end{cases} \quad (a = 10, k = 100, m = 4)$
f_{16}	Penalized2	N	[-50, 50]	0	$f_{16}(x) = 0.1 \times \left\{\sin^2(3\pi x_1) + \sum_{i=1}^{n-1}(x_i - 1)^2\left[1 + \sin^2(3\pi x_{i+1})\right] + (x_n - 1)^2\left[1 + \sin^2(2\pi x_n)\right]\right\} +$ $\sum_{i=1}^{n} u(x_i, a, k, m)$ $u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m & \text{if } x_i > a \\ 0 & \text{if } -a \le x_i \le a \\ k(-x_i - a)^m & \text{if } x_i < -a \end{cases} \quad (a = 5, k = 100, m = 4)$
f_{17}	Periodic	N	[-10, 10]	0.9	$f_{17}(\mathbf{x}) = 1 + \sum_{i=1}^{n} sin^2(x_i) - 0.1e^{(\sum_{i=1}^{n} x_i^2)}$
f_{18}	Rastrigin	S	[-5.12, 5.12]	0	$f_{18}(x) = 10n + \sum_{i=1}^{n}(x_i^2 - 10cos(2\pi x_i))$
f_{19}	Schwefel	S	[-500, 500]	0	$f_{19}(x) = 418.9829d - \sum_{i=1}^{n} x_i sin(\sqrt{\lvert x_i \rvert})$
f_{20}	Shubert	N	[-10, 10]	-186.7309	$f_{20}(\mathbf{x}) = \prod_{i=1}^{n}\left(\sum_{j=1}^{5} cos((j+1)x_i + j)\right)$

Table 3. Multimodal benchmark problems. Range: limits of search space, N: non-separable, S: separable.

Problem	Name	Type	Range	Minimum	Definition
f_{21}	Shifted Sphere Function	S	[-100, 100]	-450	$f_{21}(z)=\sum_{i=1}^{n} z_i^2 + f\_bias,\ \ z=x-o$
f_{22}	Shifted Schwefel	N	[-100, 100]	-450	$f_{22}(z)=\sum_{i=1}^{n}(\sum_{j=1}^{i} z_j)^2 + f\_bias,\ \ z=x-o$
f_{23}	Shifted Rosenbrock's	N	[-100, 100]	390	$f_{23}(z)=\sum_{i=1}^{n-1}\left((z_i^2-z_{i+1})^2+(z_i-1)^2\right)+f\_bias,\ \ z=x-o+1$
f_{24}	Shifted rotated Rastrigin's	N	[-5, 5]	-330	$f_{24}(z)=\sum_{i=1}^{n}\left(z_i^2-10\cos(2\pi z_i)+10\right)+f\_bias,\ \ z=(x-o)*M$

Table 4. Shifted and rotated benchmark problems. Range: limits of search space, N: non-separable, S: separable.

5.3 Results

In experiments, the algorithms ran for 30 times for all test functions, each time using a different initial population. We test the benchmark functions in 30 and 50 dimensions to draw empirical conclusion on the performance and scalability of the algorithms. The statistical results are reported in Tables 7–14. In these tables, *min* and *mean* are respectively the minimum and the mean function values obtained by the algorithms over 30 simulation runs. *Std* indicates the standard deviation of the results, and *Succ* indicates the number of success trials over 30 simulation runs. *Succ* is defined as

$$Succ = \bigcup_{i=1}^{30} N_{Succ}\,|_{\varepsilon}\,. \tag{14}$$

where N_{Succ} denotes the number of successful trials, in which the solution is found on ε. In simulations, an algorithm found global optimum when it converges into ε tolerance and it is defined as

$$|f_{\cos t}(T_i) - f_{\cos t}(T^*)| \leq \varepsilon \tag{15}$$

where $f_{\cos t}(T_i)$ denotes the cost function value in i^{th} iteration and $f_{\cos t}(T^*)$ indicates the global optimum of the test function.

In Tables 7–14, in order to make comparison clear, the values below 10^{-16} are assumed to be 0. In Tables 7–14, symbol "n" presents the dimension of the problems. As shown in Tables 7–14, for 30-dimension problems, the CEA algorithm performed best on 26 benchmark functions. The second and third ranks belong to JDE and SADE with 24 and 22 successes, respectively. The election algorithm, CSSA, SELO, PSO2011 and CMA-ES performed best on 22, 21, 20, 20 and 19 benchmark functions, respectively. For 50-dimension problems, the CEA algorithm performed best on 20 benchmark functions and takes the first rank. The second and third ranks belong to JDE and SADE with 18 and 16 successes, respectively. The election algorithm, CSSA, SELO, PSO2011 and CMA-ES performed best on 15, 13, 14, 9 and 13 benchmark functions, respectively. From numerical simulations,

Problem	Name	Type	Range	Minimum	Definition
f_{25}	Hybrid composition function	S	[-5, 5]	120	$F_1, F_2 =$ *Rastrigin's Function* $F_3, F_4 =$ *Weierstrass Function* $F_5, F_6 =$ *Griewank's Function* $F_7, F_8 =$ *Ackley's Function* $F_9, F_{10} =$ *Sphere Function* $[\sigma_1, \sigma_2, ..., \sigma_{10}] = [1, 1, ..., 1]$ $[\lambda_1, \lambda_2, ..., \lambda_{10}] = [1, 1, 10, 10, 5/60, 5/60, 5/32, 5/32, 5/100, 5/100]$
f_{26}	Rotated hybrid comp. Fn 1	N	[-5, 5]	120	*rotated version of* f_{25} : $F_1, F_2 =$ *Rastrigin's Function* $F_3, F_4 =$ *Weierstrass Function* $F_5, F_6 =$ *Griewank's Function* $F_7, F_8 =$ *Ackley's Function* $F_9, F_{10} =$ *Sphere Function* $[\sigma_1, \sigma_2, ..., \sigma_{10}] = [1, 1, ..., 1]$ $[\lambda_1, \lambda_2, ..., \lambda_{10}] = [1, 1, 10, 10, 5/60, 5/60, 5/32, 5/32, 5/100, 5/100]$
f_{27}	Rotated hybrid comp. Fn 2	N	[-5, 5]	310	$F_1, F_2 =$ *Ackley's Function* $F_3, F_4 =$ *Rastrigin's Function* $F_5, F_6 =$ *Sphere Function* $F_7, F_8 =$ *Weierstrass Function* $F_9, F_{10} =$ *Griewank's Function* $[\sigma_1, \sigma_2, ..., \sigma_{10}] = [1, 2, 1.5, 1.5, 1, 1, 1.5, 1.5, 2, 2]$ $[\lambda_1, \lambda_2, ..., \lambda_{10}] = [2*5/32, 5/32, 2*1, 1, 2*5/100, 5/100, 2*10, 10, 2*5/60, 5/60]$
f_{28}	Rotated hybrid comp. Fn 4	N	[-5, 5]	-330	$F_1 =$ *Weierstrass Function* $F_2 =$ *Rotated Expanded Scaffer's Function* $F_3 =$ *F8F2 Function* $F_4 =$ *Ackley's Function* $F_5 =$ *Rastrigin's Function* $F_6 =$ *Griewank's Function* $F_7 =$ *Non-Continuous Expanded Scaffer's Function* $F_8 =$ *Non-Continuous Rastrigin'sFunction* $F_9 =$ *High Conditioned Elliptic Function* $F_{10} =$ *Sphere Function with Noise in Fitness* $[\sigma_1, \sigma_2, ..., \sigma_{10}] = [1, 2, 1.5, 1.5, 1, 1, 1.5, 1.5, 2, 2]$ $[\lambda_1, \lambda_2, ..., \lambda_{10}] = [2*5/32, 5/32, 2*1, 1, 2*5/100, 5/100, 2*10, 10, 2*5/60, 5/60]$

Table 5. Hybrid benchmark problems. Range: limits of search space, N: non-separable, S: separable.

it is obvious that all algorithms have almost consistent behavior on all benchmark functions. The solution quality and convergence accuracy obtained on most test functions using the CEA in 30 independent simulation runs are almost exceeding or matching the best performance obtained by other algorithms. This testifies that the CEA has better stability behavior on most test functions rather than other algorithms.

CEA outperforms all compared algorithms on the unimodal benchmark functions in terms of the statistical test. The performance of CEA in solving multimodal benchmark problems is superior, and it generates best results in terms of *min* and *mean* values in solving 30 and 50 dimension benchmark functions. The worst results belong to CSSA, SELO and PSO2011 in solving 30 and 50 dimension multimodal benchmark functions. When solving shifted and rotated benchmark functions, CEA generally performs very well in 30 dimension benchmark functions, however, it does

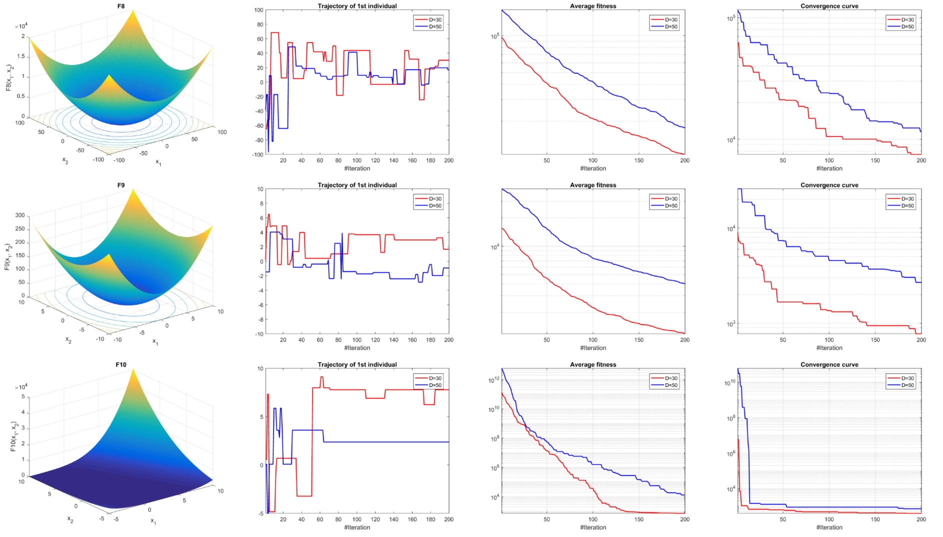
Algorithm	Control parameters
CMA-ES	$\sigma = 0.25$, $\mu = \left\lfloor \frac{4+\lfloor 2.\log(N)\rfloor}{2} \right\rfloor$
SADE	$F \sim N(0.5, 0.3)$ $CR \sim n(CR_m, 0.1)$ $c = 0.1$ $p = 0.05$
JDE	$f_{initial} = 0.5$, $CR_{initial} = 0.90$, $\tau_1 = 0.1$, $\tau_2 = 0.1$
PSO2011	$C_1 = 1.80$ $C_2 = 1.80$ ω=0.5+(1-rand)
Election algorithm	$N_C = 0.7 \times P_{size}$, $N_V = P_{size} - N_C$, Coalition rate=0.2, X_s =0.30
SELO	$P = 2$, $O = 3$, $r_p = 0.999$, $r_k = 0.1$ $follow\_prob\_factor\_ownparent = 0.999$ $follow\_prob\_factor\_otherkids = 0.9991$ $r = 0.95000$ *to* 0.99995
CSSA	$c_1 = 2e^{-\left(\frac{4l}{L}\right)^2}$, c_2, $c_3 \in [0,1]$
CEA	$N_C = 0.7 \times P_{size}$, $N_V = P_{size} - N_C$, Coalition rate=0.2, X_s =0.30 $\mu = 0.10$

Table 6. Control parameters of the algorithms used in the tests

not perform well in 50 dimension functions. SADE performs very well in solving 50 dimension shifted and rotated benchmark functions. CEA is not as competitive in solving 50 dimension hybrid functions as it does in unimodal and multimodal benchmarks. However, a careful investigation on the *mean* values shows that the performance of CEA is encouraging. From simulation results we can see that CEA performs very well in 30-dimension hybrid functions, JDE and SADE perform well in 50 dimension hybrid functions, and PSO2011 and election algorithm report the worst results in 30 and 50 dimension hybrid functions when compared with other algorithms. The *mean* and *min* values of CMA-EA, SELO, and CSSA in solving both 30 and 50 dimension functions are close to each other and show moderate results.

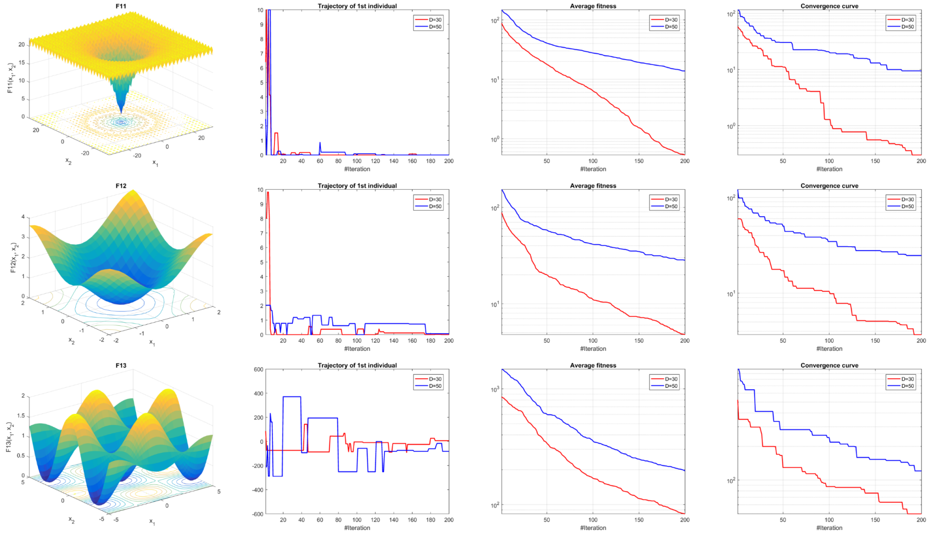
Table 15 presents the multi-problem based pairwise statistical comparison results on 30-dimension benchmark functions using the averages of the global minimum values obtained through 30 simulation runs of CEA and the comparison algorithms, based on the Wilcoxon Signed-Rank Test [26]. Table 16 presents the multi-problem based pairwise comparison results for 50-dimension benchmark functions. Multi-problem based pairwise comparisons identify which algorithm is statistically more successful in a test that includes several benchmark problems [26]. The results show that CEA was statistically more successful than other algorithms in solving the benchmark functions with a statistical significance value $\alpha = 0.05$.

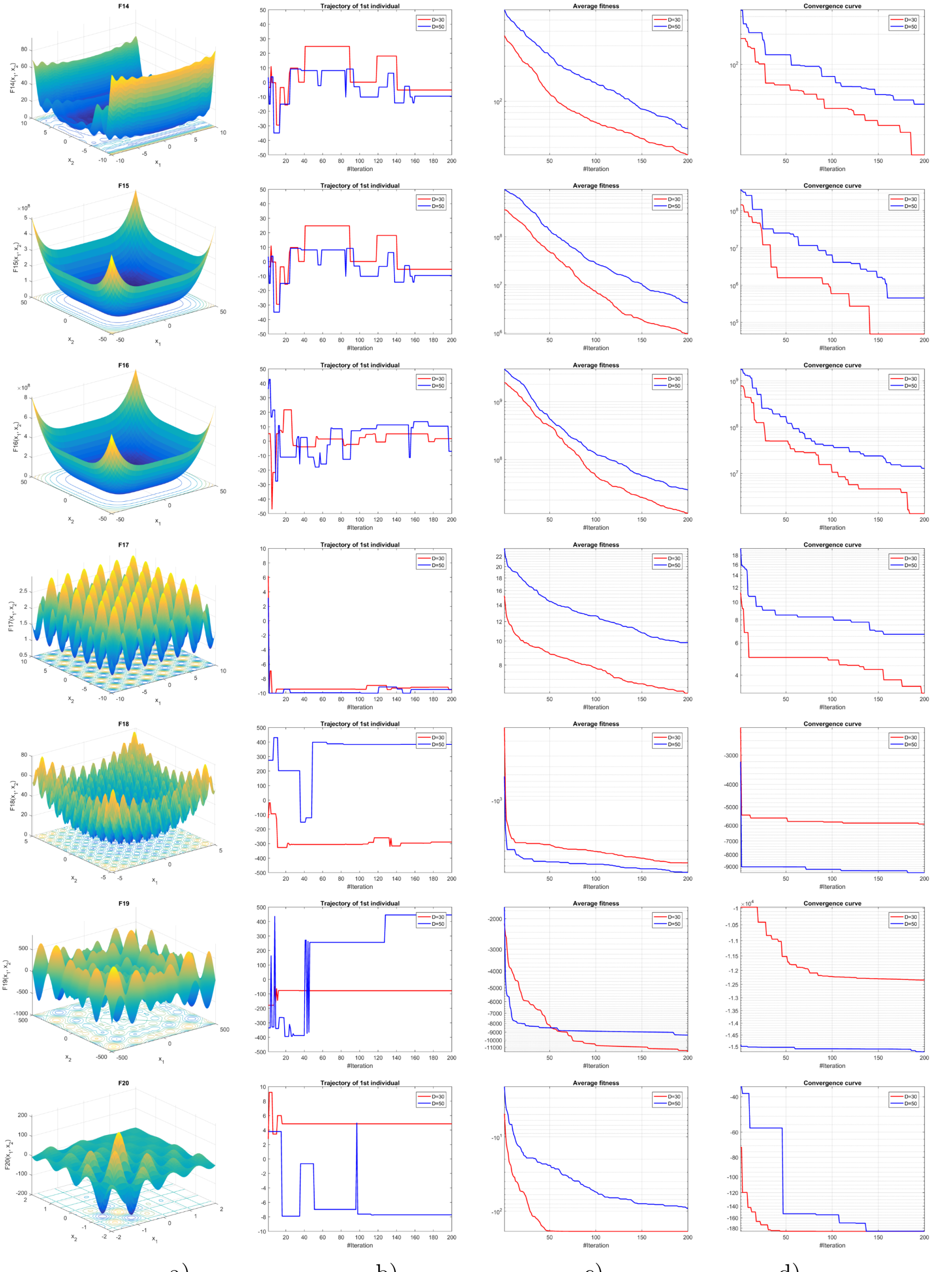
In order to observe the convergence behavior of the CEA algorithm, the convergence curve, the average fitness, and the trajectory of the first individual in its first dimension are illustrated in Figures 4, 5, 6 and 7. It should be noted that for greater clarity of plots the behavior of algorithms is shown only for 200 iterations. The second column of Figures 4, 5, 6 and 7 depicts the trajectory of the first individual in the population, in which changes of the first person in its first dimension



a) b) c) d)

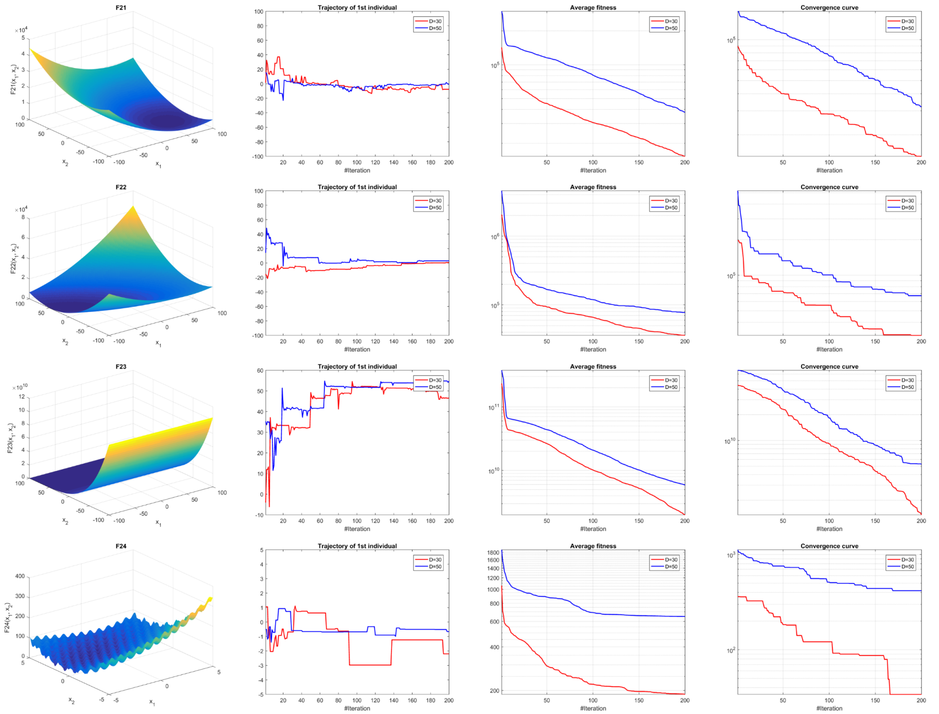
Figure 4. Results on unimodal problems, a) Graphical representations of benchmark problem, b) trajectory of the first individual in the first dimension, c) the average fitness of individuals, and d) the convergence curve of CEA algorithm





a) b) c) d)

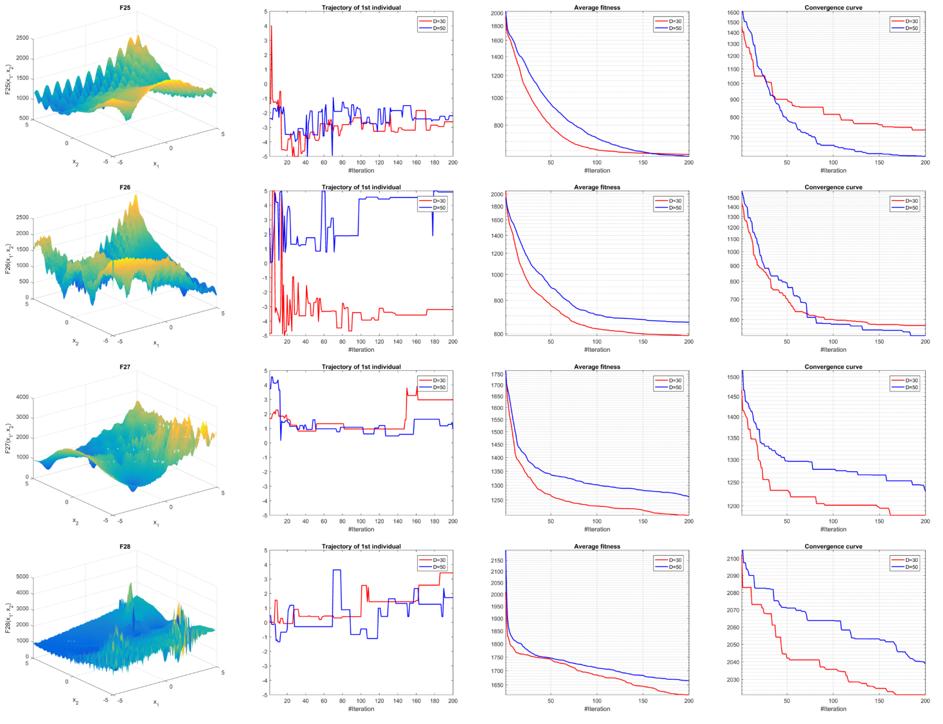
Figure 5. Results on multimodal problems, a) Graphical representations of benchmark problem, b) trajectory of the first individual in the first dimension, c) the average fitness of individuals, and d) the convergence curve of CEA algorithm



a) b) c) d)

Figure 6. Results on shifted and rotated problems, a) Graphical representations of benchmark problem, b) trajectory of the first individual in the first dimension, c) the average fitness of individuals, and d) the convergence curve of CEA algorithm

can be observed. It can be seen that there are abrupt changes in the initial steps of iterations. These abrupt changes are decreased gradually during the search process. This behavior guarantees that a population-based algorithm eventually converges to a point in search space [53, 56]. The third column of Figures 4, 5, 6 and 7 shows the average fitness that individuals obtain over 200 iterations. It can be observed that average fitness values are decreased gradually. From this behavior, it can be concluded that the fitness of individuals in the population improves through iterations. This is due to a proper balance between exploration and exploitation power of the CEA algorithm. The forth column of Figures 4, 5, 6 and 7 shows the convergence curve of CEA algorithm. It can be seen that the proposed CEA algorithm converges with a steady speed. This behavior shows the superiority of the CEA algorithm in terms of the stability and the performance. To sum up, the results verify the performance of the CEA algorithm in solving various benchmark problems compared to the counterpart algorithms. It can be concluded that the proposed CEA is an efficient algorithm for numerical function optimization.



a) b) c) d)

Figure 7. Results on hybrid problems, a) Graphical representations of benchmark problem, b) trajectory of the first individual in the first dimension, c) the average fitness of individuals, and d) the convergence curve of CEA algorithm

From the results we can see that time consumption of JDE costs least time on most test functions. CSSA costs the most time and it is in the last rank. Although CEA reported slightly more run time than JDE and SADE, their run times are comparable on most benchmarks. In contrast, CEA reports run times less than election algorithm, CMA-EA, CSSA, PSO2011 and SELO on most benchmark functions. While CEA reports more run time than JDE and SADE (on most benchmarks), its results are much better in terms of solution quality and finding global optima. On most test functions, CEA obtained the global optimum earlier before the total function evaluations. This is the reason that the time consumption of CEA is much better than the election algorithm. This justifies that the CEA is a powerful and robust extension of the election algorithm.

Problem	Statistics	CMA-ES	JDE	SADE	PSO2011	Election algorithm	SELO	CSSA	CEA
f_1	Min	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000
	Mean	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000
	Std	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000
	Succ	100	100	100	100	100	100	100	100
	Time	2.015	3.187	5.447	5.409	4.348	6.47	7.005	4.257
f_2	Min	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000022	0.0000000000000000
	Mean	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000846	0.0000000000000000
	Std	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000011	0.0000000000000000
	Succ	100	100	100	100	100	100	0	100
	Time	1.751	1.221	3.418	4.266	6.632	3.400	7.204	6.833
f_3	Min	0.6666669588917279	0.6666666666666670	0.6666666666666670	0.6666666666666720	0.6429261382035644	0.9541730938494050	0.6666666835995361	0.635574103060858
	Mean	0.6666743346755102	0.6666666666666670	0.6666666666666670	0.6666666666666750	0.6484196536514408	0.9737369841168760	0.6666667989603425	0.635970446657034
	Std	0.0000079162957191	0.0000000000000002	0.0000000000000000	0.0000000000000022	0.0000019450349497	0.0054869670667257	0.0000002248904081	0.000000250446178
	Succ	0	0	0	0	0	0	0	0
	Time	22.477	23.689	13.206	26.225	29.256	33.127	41.1351	32.836
f_4	Min	0.00018812227398726	0.0000000000000000	0.0000077644005742	0.0000095067504097	0.0000000001209500	0.0000000000000000	0.0000000000000000	0.0000000000000000
	Mean	0.00153977762312255	0.0000000000000000	0.0000110725465874	0.0000130718912008	0.0000002775466712	0.0000000000000000	0.0000000000000636	0.0000000000000000
	Std	0.00167292734740937	0.0000000000000000	0.0000034063714767	0.0000014288348929	0.0000000561469137	0.0000000000000000	0.0000000000000235	0.0000000000000000
	Succ	0	100	0	0	0	100	43.33	100
	Time	55.916	18.692	21.171	57.007	55.370	16.237	47.098	40.159
f_5	Min	0.0000000000000000	0.0000000000000000	0.0001448955835246	0.0042535368984501	0.0037741791472449	0.0000000000000000	7.1440236889946900	0.0000000000000000
	Mean	0.3986623855035210	1.0630996944802500	1.2137377447007000	2.6757043114269700	22.4020614789213527	0.0000000000000000	8.8138860361911850	0.0000000000004077
	Std	1.2164328621946200	1.7930895051734300	1.8518519388285700	12.3490058210004000	7.4158314789297055	0.0000000000000000	0.06482678290314312	0.0000000000000370
	Succ	30	20	0	0	0	100	0	33.33
	Time	82.485	19.278	45.607	43.064	37.181	13.101	34.785	30.113
f_6	Min	0.5023592840073707	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000007	0.0000000000000000	0.0000000000000000
	Mean	1.7915045718386977	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000020	0.0000000000000009	0.0000000000000000	0.0000000000000000
	Std	1.9585800691758875	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000002	0.0000000000000001	0.0000000000000000	0.0000000000000000
	Succ	0	100	100	100	83.33	0	100	100
	Time	135.024	13.679	109.551	103.764	27.673	12.688	24.315	19.163
f_7	Min	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000
	Mean	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000
	Std	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000
	Succ	100	100	100	100	100	100	100	100
	Time	18.063	3.113	8.335	٧.513	6.178	4.374	14.8644	5.832
f_8	Min	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000
	Mean	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000
	Std	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000
	Succ	100	100	100	100	100	100	100	100
	Time	7.322	3.215	4.920	6.131	9.157	1.871	16.187	9.004
f_9	Min	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000
	Mean	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000
	Std	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000
	Succ	100	100	100	100	100	100	0	100
	Time	8.670	2.098	6.383	7.229	3.577	2.115	13.70	3.180
f_{10}	Min	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000
	Mean	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000
	Std	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000
	Succ	100	100	100	100	100	100	100	100
	Time	100.868	10.165	21.488	18.083	13.213	7.351	16.1914	13.210

Table 7. Results of unimodal benchmark functions, $n = 30$

Problem	Statistics	CMA-ES	JDE	SADE	PSO2011	Election algorithm	SELO	CSSA	CEA
f_1	Min	0.0000000000000000	0.0000000000000000	0.0000000000000013	0.0000000307641078	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000
	Mean	0.0000000000000000	0.0000000000000000	0.0000000000000034	0.0000000939418142	0.0000000000000000	0.0000000000000000	0.0000000000000638	0.0000000000000000
	Std	0.0000000000000000	0.0000000000000000	0.0000000000000015	0.0000000753481345	0.0000000000000000	0.0000000000000000	0.0000000000000076	0.0000000000000000
	Succ	100	100	0	0	100	100	80	100
	Time	16.749	3.433	21.516	23.638	15.348	14.471	27.201	15.451
f_2	Min	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000001656881	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000
	Mean	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000003494273	0.0000000000009314	0.0000000000000000	0.0000000000000281	0.0000000000000000
	MIN	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000002131836	0.0000000000006654	0.0000000000000000	0.0000000000000090	0.0000000000000000
	Succ	100	100	100	0	76.67	100	83.33	100
	Time	17.135	3.032	19.731	11.3352	12.346	11.400	31.616	12.007
f_3	Min	0.6666666712080868	0.6666666666666665	0.6666666666666665	0.6666670230472646	0.6666666666840324	0.8514339096029764	0.6666668308727451	0.6380227724721356
	Mean	0.6666667405927262	0.6666666666666665	0.6666666666666665	0.6666768415722471	0.6666667195171128	1.0654716657959429	0.6666668873145076	0.6347368218366791
	Std	0.0000000729527354	0.0000000000000000	0.0000000000000000	0.0000120526677103	1.2908660020147678	0.3121960564812137	0.0000000459120280	0.0001051699387589
	Succ	0	0	0	0	0	0	0	0
	Time	25.493	26.783	18.052	20.417	24. 522	41.092	47.542	23.510
f_4	Min	0.0026415268539887	0.0000000000000000	0.0000006977481945	0.0484732633771529	0.0007936812645283	0.0000992123421870	0.0006040263968728	0.0000013599298381
	Mean	0.0031817121734536	0.0000000003809886	0.0000006977481945	0.0642270033828177	0.0018456262696528	0.0001194704095890	0.0006618000863430	0.0000019177635714
	Std	0.0004317292482951	0.0000000000300807	0.0000000000000000	0.0101406354957466	0.0006990594195686	0.0000256898252446	0.0000460410613896	0.0000003640723976
	Succ	0	50	0	0	0	0	0	0
	Time	64.994	28.158	58.800	53.746	58.740	33.306	59.220	54.001
f_5	Min	0.1618363509246701	0.0000000000000000	0.3475981703340429	10.1962821712599410	0.0799738768514800	0.0000002980396757	0.0006544015026803	0.0000000000000000
	Mean	64.2459213107620140	0.0000031909510204	62.6637216523635490	10.3574003379587990	23.9314903263880900	0.0797391612124302	0.0006069684030456	0.0011324151702647
	Std	110.9023700345396900	0.0000063608639448	51.4806879277399430	0.1816473515536368	36.3906215416394900	0.0594635542752125	0.0000115736444337	0.0019614006101603
	Succ	0	0	0	0	0	0	0	20
	Time	91.766	20.545	51.487	43.668	41.702	35.021	49.658	39.849
f_6	Min	0.0960251162413680	0.0000000000000000	0.0000000000000000	0.0764595673918435	0.0000005525442833	0.0000000023703930	0.0000000000000000	0.0000000000000000
	Mean	0.2519480590461070	0.0000000000000000	0.0000000000000000	0.0990401508526359	0.0000046955977214	0.0000006205994047	0.0000000000000651	0.0000178225610856
	Std	0.1393738482498040	0.0000000000000000	0.0000000000000000	0.0179672572699185	0.0000055169568637	0.0000009040824291	0.0000000000000068	0.0000356451221712
	Succ	0	100	100	0	0	0	80	56.67
	Time	251.166	34.821	163.252	102.394	51.794	51.721	69.605	49.383
f_7	Min	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000
	Mean	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000
	Std	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000
	Succ	100	100	100	100	100	100	100	100
	Time	31.257	7.013	12.227	13.728	10. 634	4.374	21.449	10.530
f_8	Min	0.0000000000000000	0.0000000000000000	0.0000000000000000	0. 0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000
	Mean	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000019808005301	0.0000004349891225	0.0000000014838244	0.0000000000000000	0.0000000000000000
	Std	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000007641013227	0.0000005546172377	0.0000000012109497	0.0000000000000000	0.0000000000000000
	Succ	100	100	100	76.67	76.67	80	100	100
	Time	20.286	4.972	8.481	14.138	11.183	16.205	19.281	11.196
f_9	Min	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000
	Mean	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000
	Std	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000
	Succ	100	100	100	100	100	100	100	100
	Time	21.634	3.188	10.850	9.110	8.201	6.462	50.794	27.818
f_{10}	Min	624.5630179244759600	0.0000000000000000	0.0000000000000000	0.0453695392950734	0.0000000000000000	0.0004971972185198	0.0000000000000000	0.0000000000000000
	Mean	963.9116925353591800	0.0000000000000000	0.0000000000000000	0.0946941941778695	0.0000000000003423	0.0017045282408875	0.0000000000000000	0.0000000000000000
	Std	201.1185662003226200	0.0000000000000000	0.0000000000000000	0.0359721171318677	0.0000000000003651	0.0011890090919424	0.0000000000000000	0.0000000000000000
	Succ	0	100	100	0	80	0	0	100
	Time	202.417	22.730	69.996	23.198	22.748	45.951	19.699	19.240

Table 8. Results of unimodal benchmark functions, $n = 50$

Problem	Statistics	CMA-ES	JDE	SADE	PSO2011	Election algorithm	SELO	CSSA	CEA
f_{11}	Min	0.0000000000000027	0.0000000000030836	0.0000000000000027	0.0000000000000080	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000
	Mean	0.0000000000000027	0.0000000004992996	0.0000000000000027	1.5214322973725000	0.0000000000000000	0.0000000000000000	0.0000008834796247	0.0000000000000000
	Std	0.0000000000000000	0.0000000002573112	0.0000000000000000	0.6617570384662600	0.0000000000000000	0.0000000000000000	0.0000005120149712	0.0000000000000000
	Succ	0	0	0	0	100	100	0	100
	Time	30.287	25.136	10.428	23.039	9.250	5.674	28.616	9.222
f_{12}	Min	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000
	Mean	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000247	0.0000000000005810	0.0000000000000000
	Std	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000015	0.0000000000000339	0.0000000000000000
	Succ	100	100	100	100	100	73.34	36.67	100
	Time	15.604	6.140	56.738	29.248	26.877	27.328	51.705	23.751
F_{13}	Min	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000274060	0.0000000000000000
	Mean	0.0000000000000000	0.0000000000000000	0.0226359326967139	0.0068943694819713	0.0000000000000000	0.0000000000000000	0.0000140271987100	0.0000000000000000
	Std	0.0000000000000000	0.0000000000000000	0.0283874287215679	0.0080565201649587	0.0000000000000000	0.0000000000000000	0.0000041779936673	0.0000000000000000
	Succ	100	100	36.66	50	100	100	0	100
	Time	2.647	6.914	25.858	23.895	9.1919	7.940	55.4761	7.16962
f_{14}	Min	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000
	Mean	0.0000000000000000	0.0020185116261490	0.0000000000000000	0.0000000000001254	0.0000000000000000	0.0000000000000012	0.0000000000000000	0.0000000000000000
	Std	0.0000000000000000	0.0077448684015362	0.0000000000000000	0.00000000000001503	0.0000000000000000	0.0000000000000001	0.0000000000000000	0.0000000000000000
	Succ	100	100	100	83.34	100	93.34	100	100
	Time	۵.062	6.692	7.886	9.441	4.685	9.492	26.811	4.116
f_{15}	Min	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000
	Mean	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000043	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000
	Std	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000015	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000
	Succ	100	100	100	90	100	100	100	100
	Time	5.851	9.492	15.992	19.600	8.441	18.075	48.514	8.411
f_{16}	Min	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000
	Mean	0.0003662455278628	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000
	Std	0.0020060093719584	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000
	Succ	33.33	100	100	100	100	100	100	100
	Time	6.158	1.969	4.544	12.507	11.715	9.537	31.668	12.324
f_{17}	Min	7.5330185062452992	1.0000000000000002	1.000000000000004	1.0000000000000002	1.0000000000489453	0.9000000000001315	1.0000000000074536	0.9000000000000000
	Mean	8.0447358655943226	1.0000000000000004	1.1305510725002295	1.0000000000000042	1.0000000000591651	1.0000000000055178	1.0000000000102574	0.9200000000175261
	Std	0.1630535915023040	0.0000000000000002	0.0107698043735300	0.0000000000000011	0.0000000000067066	0.0000000000001234	0.0000000000022537	0.0399999999912369
	Succ	0	0	0	0	0	0	0	43.33
	Time	76.081	24.002	29.688	48.170	36.198	73.124	48.5810	23.909
f_{18}	Min	29.8487565993415000	0.0000000000000000	0.0000000000000000	12.9344677422129000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000
	Mean	95.9799861204982000	0.0000000000000000	0.8622978494808570	25.6367602258676000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000
	Std	56.6919245985100000	0.0000000000000000	0.9323785263847000	8.2943512684216700	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000
	Succ	0	100	26.67	0	100	100	100	100
	Time	27.40	3.635	13.594	16.023	7.988	3.941	15.812	7.187
f_{19}	Min	-8340.0386911070600000	-12569.4866181730000000	-12569.4866181730000000	-8912.8855854978200000	-12569.4866181730140000	-0.0325083488969540	-8542.3106048357640000	-12569.4866181730140000
	Mean	-6835.1836730901400000	-12304.9743375341000000	-12549.7468957373000000	-7684.6104757783800000	-12036.1119932232430000	-0.3402784042291390	-7796.5971553930576000	-12166.6227102694010000
	Std	750.7338055436110000	221.4322514436480000	44.8939348779747000	745.3954005014180000	923.8319498809704600	3.2212919091274600	658.0863939861569600	805.7278158072265300
	Succ	0	0	0	0	0	0	0	0
	Time	6.174	10.315	34.383	30.427	15.5106	17.419	20.797	11.143
f_{20}	Min	-186.7309088310240000	-186.7309088310240000	-186.7309088310240000	-186.7309088310240000	-186.7309088310240000	-186.7363874875390000	-186.7309088310240000	-186.7309088310240000
	Mean	-81.5609772893002000	-186.7309088310240000	-186.7309088310239000	-186.7309073569880000	-186.7309088310239200	-186.7153981691330000	-186.7309088310238700	-186.7309088310239000
	Std	66.4508342743478000	0.0000000000000388	0.0000000000000377	0.0000046401472660	0.0000000000000027826	0.0190762312882078	0.0000000000000180	0.0000000000000000393
	Succ	13.34	63.34	66.67	30	60	0	66.67	70
	Time	25.225	8.213	27.109	19.770	31.2819	25.147	69.338	26.335

Table 9. Results of multimodal benchmark functions, $n = 30$

Problem	Statistics	CMA-ES	JDE	SADE	PSO2011	Election algorithm	SELO	CSSA	CEA
f_{11}	Min	0.0000000000000027	0.0000000000000027	0.0000000000000027	0.0000000000004000	0.0000000000000000	0.0000000000000000	0.0000007266174480	0.0000000000000000
	Mean	0.0000000000000027	0.0000000000000027	0.0000000000000027	1.7266174422716470	0.0000000000000000	0.0000000000000000	0.0000051230641084	0.0000000000000000
	Std	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.946515355 044351	0.0000000000000000	0.0000000000000000	0.0000001550003205	0.0000000000000000
	Succ	0	0	0	0	100	100	0	100
	Time	37.402	27.267	25.373	37.110	11.041	9.704	95.117	10.984
f_{12}	Min	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000
	Mean	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000001566247	0.0000000528283220	0.0000000000000000
	Std	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000375115	0.0000000004796087	0.0000000000000000
	Succ	100	100	100	100	100	50	30	100
	Time	33.568	8.010	53.258	41.066	32.267	38.115	57.338	25.170
F_{13}	Min	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000051235828	0.0000000000000000
	Mean	0.0000000000000000	0.0000000000000000	0.0009845014593703	0.0068943694819713	0.0000000000000000	0.0000000000000000	0.0018490161387324	0.0000000000000000
	Std	0.0000000000000000	0.0000000000000000	0.00079649632056923	0.0080565201649587	0.0000000000000000	0.0000000000000000	0.0032025787311004	0.0000000000000000
	Succ	100	100	86.67	50	100	100	0	100
	Time	17.862	5.355	24.6358	22.533	12.665	9.385	65.850	11.004
f_{14}	Min	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0007913556231509	0.0000000000000000
	Mean	0.0000000000000000	0.0000000000000000	0.4733646979405994	0.3133488764575432	0.0000000000000000	0.0000089528250416	0.0013102112517912	0.0000000000000000
	Std	0.0000000000000000	0.0000000000000000	0.3741896258335322	0.1060956269188929	0.0000000000000000	0.0000000633060329	0.0005322648501157	0.0000000000000000
	Succ	100	100	30	20	100	40	0	100
	Time	20.321	6.702	29.764	28.003	9.344	39.127	43.561	9.760
f_{15}	Min	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000018	0.0000000000000000	0.0000000000000000
	Mean	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.1278728062391630	0.0000000000000000	0.0000642528164196	0.0000000000000003	0.0000000000000000
	Std	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.2772792346028400	0.0000000000000000	0.0000151668177694	0.0000000000000002	0.0000000000000000
	Succ	100	100	100	16.66	100	50	80	100
	Time	20.551	4.0768	8.2708	39.555	9.565	34.087	51.196	9.402
f_{16}	Min	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0234368584375104	0.0000000000000000
	Mean	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0043949463343535	0.0000000000000000	0.0000000000000000	0.9453148964751007	0.0000000000000000
	Std	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0054747064090174	0.0000000000000000	0.0000000000000000	0.1531241868389090	0.0000000000000000
	Succ	100	100	100	16.66	100	100	0	100
	Time	18.993	6.201	9.203	16.137	16.017	17.106	35.648	15.297
f_{17}	Min	14.0566700961880930	1.0000000000000004	1.0000000002066967	1.3795783622590090	1.0000000000369984	1.0000000000057578	0.9000000000000000	0.9000000000000000
	Mean	14.3884549392425620	1.0000000000000011	1.0000006030324166	1.4181834652710708	1.0000000000449019	1.0000000000097171	0.9750000000078725	0.9600000000000000
	Std	0.3017442249329344	0.0000000000000009	0.0000000374858280	0.0254098620222145	0.0000000000053155	0.0000000000025909	0.0433012701937671	0.0489897948556636
	Succ	0	0	0	0	0	0	23.33	50
	Time	252.480	56.117	108.077	62.924	61.730	59.648	65.159	57.322
f_{18}	Min	15.9193449134927500	0.0000000000000000	0.0000000000000000	65.6672473922326960	0.0000000000000000	0.0000000000000000	0.0000000000084371	0.00000000000000000
	Mean	224.9814655516387900	0.0000000000000000	0.2653902873322949	93.0285078955684380	0.9958240635327371	0.0000000000000000	0.0000000000140816	0.00000000000009949
	Std	121.2874373727220700	0.0000000000000000	0.4247444940971572	15.9890829391755500	1. 4026979638646450	0.0000000000000000	0.0000000000016255	0.00000000000070354
	Succ	0	100	43.33	0	20	100	0	70
	Time	191.468	4.922	16.738	55.0321	15.110	14.814	37.8123	15.817
f_{19}	Min	-Inf	-20949.1443632022420000	-20949.1443636216720000	-14342.5691261116770000	-20949.1443636216720000	-19830.1777064995950000	-13337.3783150455510000	-20949.1443630272190000
	Mean	-Inf	-20949.1443620160350000	-20949.1443636216720000	-13770.2818275667050000	-17612.3302345816880000	-18832.9353757876050000	-13019.9077731192970000	-20949.1443612843000000
	Std	NaN	0.0000011287140530	0.0000000000000000	340.9143617460632100	2737.4387810085359000	881.1129403390057200	265.0087423288549100	0.0000029487969428
	Succ	0	0	0	0	0	0	0	0
	Time	-	20.422	22.435s	20.899	21.750	25.132	58.4874	18.454
f_{20}	Min	-186.7261812733892400	-186.7309088310239800	-186.7309088310239800	-186.7309088308585700	-186.7309088309020800	-186.7309088310239800	-186.7309088310240000	-186.7309088310240000
	Mean	-186.6819266547511200	186.7309088310239800	-186.7309088310239200	-186.7309083581201300	-186.7309086707775200	-186.7309088310239500	-186.7309088310237800	-186.7309065904519200
	Std	0.0483046109651153	0.0000000000000284	0.0000000000000318	0.0000008659591989	0.0000003137304075	0.0000000000000376	0.0000000000000651	0.0000007440998569
	Succ	0	0	0	0	0	0	46.67	53.33
	Time	275.1913	29.203	37.847	22.3088	34.366	41.954	36.791	29.075

Table 10. Results of multimodal benchmark functions, $n = 50$

Problem	Statistics	CMA-ES	JDE	SADE	PSO2011	Election algorithm	SELO	CSSA	CEA
f_{21}	Min	-450.0000000000000000	-450.0000000000000000	-450.0000000000000000	-450.0000000000000000	-450.0000000000000000	-450.0000000000000000	-450.0000000000000000	-450.0000000000000000
	Mean	-450.0000000000000000	-450.0000000000000000	-450.0000000000000000	-450.0000000000000000	-450.0000000000000000	-450.0000000000000000	-450.0000000000000000	-450.0000000000000000
	Std	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000
	Succ	100	100	100	100	100	100	100	100
	Time	83.144	11.166	39.155	38.136	43.544	37.250	63.972	44.088
f_{22}	Min	-450.0000000000000000	-450.0000000000000000	-450.0000000000000000	-450.0000000000000000	-450.0000000000000000	-450.0000000000000000	-450.0000000000000000	-450.0000000000000000
	Mean	-450.0000000000000000	-450.0000000000000000	-450.0000000000000000	-450.0000000000000000	-450.0000000000000000	-441.8724485886073130	-450.0000000000000000	-450.0000000000000000
	Std	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	12.0594053705881000	0.0000000000000000	0.0000000000000000
	Succ	100	100	100	100	100	50	100	100
	Time	21.701	31.067	65.375	59.178	37.922	62.197	57.359	33.430
f_{23}	Min	390.0000000000000000	390.0000000000000000	390.0000000000000000	390.0000000032170240	390.0000000000000000	411.2548416301633000	390.0000000000000000	390.0000000000000000
	Mean	390.5315438816460000	390.0000000000001100	390.0000000000011900	398.2000147834449103	392.5883754233423700	414.6731243841997500	413.2364746591623500	390.3081451255193200
	Std	2.50007412834795167	0.0000000000000568	0.0000000000018808	16.2217059831472681	3.1053936255249658	2.4370774444214609	29.7345673904860000	0.47905185297883029
	Succ	26.67	50	30	0	26.67	0	30	40
	Time	59.432	28.733	57.149	77.319	62.439	102.734	116.892	60.739
f_{24}	Min	-187.9511478720692100	-328.0100818858134100	-326.0201637716268100	-327.7654123597851270	-312.8425436863587900	-283.7415542110191800	-311.789711627220061	-310.7873572616098800
	Mean	-170.8132987651011100	-326.7663830644468100	-324.0302481763498000	-323.9710235600048743	-294.7985918252179500	-157.3753555505669500	-297.620443705256010	-305.2082580888827000
	Std	10.1551136331903620	1.2924897287523103	1.2185689457394560	4.00000513791587136	12.2729097988707100	37.1250046662017720	32. 4838827283423496	7.3223833415277388
	Succ	0	0	0	0	0	0	0	0
	Time	193.0771	79.196	81.979	86.221	67.836	86.720	98.316	70.469

Table 11. Results of shifted and rotated benchmark functions, $n = 30$

Problem	Statistics	CMA-ES	JDE	SADE	PSO2011	Election algorithm	SELO	CSSA	CEA
f_{21}	Min	-450.0000000000000000	-450.0000000000000000	-450.0000000000000000	-450.0000000000000000	-450.0000000000000000	-450.0000000000000000	-450.0000000000000000	-450.0000000000000000
	Mean	-450.0000000000000000	-449.9999999999999400	-450.0000000000000000	-450.0000000000000000	-450.0000000000000000	-450.0000000000000000	-450.0000000000000000	-450.0000000000000000
	Std	0.0000000000000000	0.0000000000000508	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000
	Succ	100	50	100	100	100	100	100	**100**
	Time	134.515	34.837	40.705	44.803	46.157	46.330	75.084	47.028
f_{22}	Min	-450.0000000000000000	-449.9999999999996600	-450.0000000000000000	-450.0000000000000000	-450.0000000000000000	-450.0000000000000000	-449.2047023577194400	-450.0000000000000000
	Mean	-450.0000000000000000	-449.9999999999995500	-450.0000000000000000	-450.0000000000000000	-450.0000000000000000	-449.9923244886244900	-446.9325426695196500	-450.0000000000000000
	Std	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0044382733553957	3.3286174731900031	0.0000000000000000
	Succ	100	0	100	100	100	50	0	**100**
	Time	153.189	36.635	65.375	110.178	83. 923	70.521	102.853	67.147
f_{23}	Min	923.3122301231220500	390.2651577762746900	390.0433169643209200	390.0000000032170240	390.1349684072680000	406.1913933149286300	390.0000000774123800	390.0000682074815500
	Mean	6004.1402501951288000	390.2651577762746900	425.3330809945244400	398.2000147834449103	395.7864367244400000	458.9883748242529000	390.8192072680831200	391.7998112680409700
	Std	5594.7482946023929000	0.0000000000000000	34.9323508668616260	16.2217059831472681	7.3178607152460000	27.3512160883286800	1.5914652405331551	1.7962980906277657
	Succ	0	0	0	0	0	0	0	0
	Time	195.009	152.979	116.520	144.319	66.613	95.125	120.633	60.548
f_{24}	Min	-262.0000858641211600	-290.2016528297256200	-223.5396434040115400	-215.5799000531476400	-239.1246610103919700	-224.5346074989227500	-288.4640698301597000	-292.7043735974306200
	Mean	-236.2931389112091000	-281.3981485447598100	-207.2224344191213800	-194.4871539557644800	-230.0882309073202200	-206.8741753507984000	-235.2322820353835500	-271.0274687832123800
	Std	15.4167083463846360	5.3345602587061283	14.3714838213930720	20.3456873003582500	5.2380481185273311	16.5798203539693250	28.3769846149553120	13.3149257435571790
	Succ	0	0	0	0	0	0	0	0
	Time	176.420	120.380	101.025	89.199	115.902	132.213	246.338	107.349

Table 12. Results of shifted and rotated benchmark functions, $n = 50$

Problem	Statistics	CMA-ES	JDE	SADE	PSO2011	Election algorithm	SELO	CSSA	CEA
f_{25}	Min	234.2514754322145003	120.0000000000000000	120.0000000000000000	120.0000000000000000	120.0000000000000000	120.0000000000000000	120.0000000000000000	120.0000000000000000
	Mean	455.0851123744125896	161.5045411507748800	231.1247852578964788	415.4627658246898124	251.2678634156410020	301.8803634401870700	247.1679437402640000	220.4566396417482100
	Std	167.1554863214578955	40.8123547862124458	152.1047684571247851	146.2245789214568549	157.3938282341687970	182.2785372619515600	163.8506305541810000	134.5789554563156700
	Succ	0	33.34	10	16.67	40	10	20	40
	Time	158.217	195.247	156.294	202.541	175.519	168.153	315.110	170.467
f_{26}	Min	120.0000000000000000	221.9559596030777900	213.8866174069701500	178.1500148357924853	238.3285405884752200	216.4603102871090000	138.1239330780676900	230.4697363886152700
	Mean	221.6635496595235100	228.3093455824848900	216.0670953318125700	230.0043894117463852	253.7391338926102300	328.8991571751680000	215.0891175978302400	250.0590278771736600
	Std	40.4793070004110830	4.4197947468679075	2.7755155824340010	16.6615785146794134 0	17.0097061443176365	169.3910734871141100	110.4370379004781702	17.1082798514738560
	Succ	20	0	0	0	0	0	0	0
	Time	747.772	120.440	113.098	156.571	129.621	106.391	226.638	113.071
f_{27}	Min	310.0000000000000000	310.0000000000000000	310.0000000000000000	310.0000000000000000	310.0000000000000000	310.0000000000000000	310.0000000000000000	310.0000000000000000
	Mean	855.5086440559429100	310.0000000000000000	810.0000154320209700	539.9127648531200066	310.0000000000000000	760.1441063457674500	921.4554355013506200	310.0000000000000000
	Std	166.7753968101395050	0.0000000000000000	120.4615734891640092	139.4157431958731500	0.0000000000000000	237.9472146523420480	203.5667647189327747	0.0000000000000000
	Succ	20	100	30	20	100	20	16.67	100
	Time	400.744	180.943	250.709	248.043	195.573	174.396	328.6237	180.042
f_{28}	Min	460.0000000000000000	460.0000000000000000	460.0000000000000000	460.0000000000000000	460.0000000000000000	460.0000000000000000	460.0000000000000000	460.0000000000000000
	Mean	639.1997430436929300	496.3517246697841250	486.0000000301876248	470.9388147562189 7413	460.0000000000000000	501.1324892514786200	610.3068519876720800	460.0000000000000000
	Std	115.4619998211036105	51.1203578914876000	34.6021574698525521	45.0217893158004 1978	0.0000000000000000	130.3485147618954797	150.2745692476307200	0.0000000000000000
	Succ	33.34	56.67	63.34	20	100	50	23.34	100
	Time	251.472	95.099	183.511	163.150	159.221	145.307	215.507	158.921

Table 13. Results of hybrid benchmark functions, $n = 30$

Problem	Statistics	CMA-ES	JDE	SADE	PSO2011	Election algorithm	SELO	CSSA	CEA
f_{25}	Min	396.6452204913950900	320.000000000000000	421.0505168707150000	439.9360251174790600	443.0833104620444400	330.5889241782028900	358.1087818436275300	365.7198427286517100
	Mean	480.3266429050074700	422.4652412915387500	496.0041304732940300	487.3270263606730700	515.7121551751708900	482.1177848356405800	374.5817626615117000	638.1043872977845700
	Std	91.3460815719953100	100.3165747092249400	43.2914263451065300	39.9836893568482590	41.9448125432753680	75.7644303287188450	15.6650035501275000	200.0575903312088700
	Succ	0	0	0	0	0	0	0	0
	Time	491.786	298.295	319.425	307.517	350.546	282.164	361.275	356.520
f_{26}	Min	120.0000000000000000	217.1900815567871900	199.9382210829309100	278.2154218390603000	190.8414908573395500	184.0157761153815800	149.1922042052661900	192.7732803163116300
	Mean	196.7110237013338900	252.6683601225867900	223.9699491162265000	327.0597704682400000	206.0035312273601000	198.5561626310519000	230.9178687302762700	200.0198512342537400
	Std	57.1921651574837780	32.3475224810199489	35.5713239484792610	52.5087126644775070	9.0692872306153571	12.6271827756316280	50.3369274848248230	6.4160931595084012
	Succ	10	0	0	0	0	0	0	0
	Time	931.186	185.062	159.9185	154.903	209.122	171.186	328.580	198.559
f_{27}	Min	850.4272847542918000	896.3547537249504600	873.7341022604559800	866.9770245102914700	851.3122051250039700	850.7138279453589600	855.0006511545613000	852.2913959787829300
	Mean	987.1374598307908200	905.2151711585196400	888.0783375694557000	894.4624097928066200	916.0776838272911400	915.4467335913683400	910.1927459268479100	912.8355132527331100
	Std	157.9085076379647732	4.5521655742118154	12.6807771772059020	16.3342488132452140	39.9225774311974000	55.2168825407607000	187.0135958016884677	29.1706442006170550
	Succ	0	0	0	0	0	0	0	0
	Time	582.924	240.531	290.079	285.772	331.805	317.105	343.0724	316.988
f_{28}	Min	460.1038377611588400	460.0000000000033000	460.0000000000001100	460.0000000000034100	460.1446074981029000	460.0000000000070500	460.0000000000005630	460.0000000000006900
	Mean	533.3414369696136000	460.0000000000033500	482.3554625773845000	507.3288440371169400	608.3316183481684000	664.4400697095233000	660.6319018127858300	556.5771419380732800
	Std	50.9414908921000604	0.0000000000000284	19.9666695102418400	126.3417283082863500	111.4992852066428700	109.4638592242058600	201.7638036255714800	88.3115525305439280
	Succ	0	0	0	0	0	0	0	0
	Time	322.232	115.933	203.169	187.480	183.390	173.537	239.971	175.477

Table 14. Results of hybrid benchmark functions, $n = 50$

Comparison	T+	T-	*p*-value	Winner
CEA vs. CMA-ES	39	6	0.02852	CEA
CEA vs. JDE	7	3	-*	CEA
CEA vs. SADE	11	10	-	CEA
CEA vs. PSO2011	30	15	0.2040	CEA
CEA vs. Election algorithm	11	4	-	CEA
CEA vs. SELO	17	4	-	CEA
CEA vs. CSSA	13	8	-	CEA

* Symbol "-" means that it is not possible to calculate an accurate *p*-value

Table 15. Multi-problem based statistical pairwise comparison of CEA and comparison algorithms using two-sided Wilcoxon Signed-Rank test ($\alpha = 0.05$), $n = 30$

Comparison	T+	T-	*p*-value	Winner
CEA vs. CMA-ES	101	19	0.0114	CEA
CEA vs. JDE	33	12	0.01928	CEA
CEA vs. SADE	52	3	0.00298	CEA
CEA vs. PSO2011	103	2	0.00028	CEA
CEA vs. Election algorithm	57	21	0.08726	CEA
CEA vs. SELO	53	25	0.0466	CEA
CEA vs. CSSA	77	28	0.06876	CEA

Table 16. Multi-problem based statistical pairwise comparison of CEA and comparison algorithms using two-sided Wilcoxon Signed-Rank test ($\alpha = 0.05$), $n = 50$

6 CONCLUSION

This paper presents a Chaotic Election Algorithm (CEA) to improve the original election algorithm. The CEA enhances the election algorithm threefold:

1. modifying party formation phase,
2. introducing migration operator, and
3. introducing a chaotic positive advertisement.

CEA by modifying the party formation phase through eliminating the Euclidean distance computation from the process increases the speed of the algorithm. With the migration operator, diversity in the population is maintained, what keeps the CEA away from converging too fast before exploring the entire solution space. With the new chaotic positive advertisement, the information exchanges between candidates and voters efficiently and improves the algorithm's search ability. To show the performance of the CEA algorithm, it is evaluated on 28 optimization benchmarks and compared with CMA-EA, JDE, SADE, PSO2011, SELO, CSSA and election algorithm. The results show that the proposed CEA algorithm outperforms the canonical election algorithm and other comparable counterparts in terms of solution quality and convergence speed. There remain several points to improve our research. First, the CEA will be trapped in local optimums on few functions, which can be seen from simulation results on some benchmark functions. We can combine the CEA with some local search strategies or other meta-heuristics to further enhance its optimization ability. Second, we can apply the proposed CEA algorithm to solve more practical optimization problems to accurately identify its weaknesses and merits. Third, in some specific engineering applications, some components of the algorithm can be modified in order to improve the performance of the algorithm.

REFERENCES

[1] Sotoudeh-Anvari, A.—Hafezalkotob, A.: A Bibliography of Metaheuristics – Review from 2009 to 2015. International Journal of Knowledge-Based and Intelligent Engineering Systems, Vol. 22, 2018, No. 1, pp. 83–95, doi: 10.3233/KES-180376.

[2] Gogna, A.—Tayal, A.: Metaheuristics: Review and Application. Journal of Experimental and Theoretical Artificial Intelligence, Vol. 25, 2013, No. 4, pp. 503–526, doi: 10.1080/0952813X.2013.782347.

[3] Boussaïd, I.—Lepagnot, J.—Siarry, P.: A Survey on Optimization Metaheuristics. Information Sciences, Vol. 237, 2013, pp. 82–117, doi: 10.1016/j.ins.2013.02.041.

[4] Hussain, K.—Mohd Salleh, M. N.—Cheng, S.—Shi, Y.: Metaheuristic Research: A Comprehensive Survey. Artificial Intelligence Review, Vol. 52, 2019, pp. 2191–2233, doi: 10.1007/s10462-017-9605-z.

[5] Beheshti, Z.—Shamsuddin, S. M.: A Review of Population-Based Meta-Heuristic Algorithms. International Journal of Advances in Soft Computing and Its Applications, Vol. 5, 2013, No. 1, pp. 1–35.

[6] Alsalibi, B.—Venkat, I.—Subramanian, K. G.—Lufti, S. L.—De Wilde, P.: The Impact of Bio-Inspired Approaches Toward the Advancement of Face Recognition. ACM Computing Surveys (CSUR), Vol. 48, 2015, No. 1, Art. No. 5, 31 pp., doi: 10.1145/2791121.

[7] José-García, A.—Gómez-Flores, W.: Automatic Clustering Using Nature-Inspired Metaheuristics: A Survey. Applied Soft Computing, Vol. 41, 2016, pp. 192–213, doi: 10.1016/j.asoc.2015.12.001.

[8] Engelbrecht, A. P.: Computational Intelligence: An Introduction. 2nd Edition. Wiley, New York, 2007.

[9] Haupt, R. L.—Haupt, S. E.: Practical Genetic Algorithms. 2nd Edition. Wiley, New York, 2004, doi: 10.1002/0471671746.

[10] Civicioglu, P.: Backtracking Search Optimization Algorithm for Numerical Optimization Problems. Applied Mathematics and Computation, Vol. 219, 2013, No. 15, pp. 8121–8144, doi: 10.1016/j.amc.2013.02.017.

[11] Karaboga, D.—Akay, B.: A Comparative Study of Artificial Bee Colony Algorithm. Applied Mathematics and Computation, Vol. 214, 2009, No. 1, pp. 108–132, doi: 10.1016/j.amc.2009.03.090.

[12] Emami, H.—Derakhshan, F.: Election Algorithm: A New Socio-Politically Inspired Strategy. AI Communications, Vol. 28, 2015, pp. 591–603, doi: 10.3233/AIC-140652.

[13] Kennedy, J.—Eberhart, R.: Particle Swarm Optimization. Proceedings of the IEEE International Conference on Neural Networks (ICNN '95), Perth, WA, Australia, 1995, pp. 1942–1948, doi: 10.1109/ICNN.1995.488968.

[14] Ghaemi, M.—Feizi-Derakhshi, M.-R.: Forest Optimization Algorithm. Expert Systems with Applications, Vol. 41, 2014, No. 15, pp. 6676–6687, doi: 10.1016/j.eswa.2014.05.009.

[15] Shah-Hosseini, H.: Problem Solving by Intelligent Water Drops. 2007 IEEE Congress on Evolutionary Computation (CEC 2007), 2007, pp. 3226–3231, doi: 10.1109/CEC.2007.4424885.

[16] Kaveh, A.—Talatahari, S.: A Novel Heuristic Optimization Method: Charged System Search. Acta Mechanica, Vol. 213, 2010, pp. 267–289, doi: 10.1007/s00707-009-0270-4.

[17] Hatamlou, A.: Black Hole: A New Heuristic Optimization Approach for Data Clustering. Information Sciences, Vol. 222, 2013, pp. 175–184, doi: 10.1016/j.ins.2012.08.023.

[18] Tayarani, M.—Akbarzadeh, M.: Magnetic-Inspired Optimization Algorithms: Operators and Structures. Swarm and Evolutionary Computation, Vol. 19, 2014, pp. 82–101, doi: 10.1016/j.swevo.2014.06.004.

[19] Biswas, A.—Mishra, K. K.—Tiwari, S.—Misra, A. K.: Physics-Inspired Optimization Algorithms: A Survey. Journal of Optimization, Vol. 2013, 2013, Art. No. 438152, 16 pp., doi: 10.1155/2013/438152.

[20] Emami, H.—Derakhshan, F.: Integrating Fuzzy K-Means, Particle Swarm Optimization and Imperialist Competitive Algorithm for Data Clustering. Arabian Journal for Science and Engineering, Vol. 40, 2015, pp. 3545–3554, doi: 10.1007/s13369-015-1826-3.

[21] Lokhande, N. M.—Pujeri, R. V.: Novel Image Segmentation Using Particle Swarm Optimization. Proceedings of the 2018 8$^{\text{th}}$ International Conference on Biomedical Engineering and Technology (ICBET 2018), ACM, 2018, pp. 46–50, doi: 10.1145/3208955.3208962.

[22] Guerrero, M.—Montoya, F. G.—Baños, R.—Alcayde, A.—Gil, C.: Adaptive Community Detection in Complex Networks Using Genetic Algorithms. Neurocomputing, Vol. 266, 2017, pp. 101–113, doi: 10.1016/j.neucom.2017.05.029.

[23] Emami, H.—Derakhshan, F.: A New Method for Conflict Detection and Resolution in Air Traffic Management. Twenty-Sixth AAAI Conference on Artificial Intelligence @ Semantic Cities, 22–26 July 2012, Toronto, Canada, AAAI Press, pp. 37–40.

[24] Wolpert, D.—Macready, W. G.: No Free Lunch Theorems for Optimization. IEEE Transactions on Evolutionary Computation, Vol. 1, 1997, No. 1, pp. 67–82, doi: 10.1109/4235.585893.

[25] Vesterstrøm, J.—Thomsen, R.: A Comparative Study of Differential Evolution Particle Swarm Optimization and Evolutionary Algorithms on Numerical Benchmark Problems. Proceedings of the 2004 Congress on Evolutionary Computation (CEC 2004), IEEE, Vol. 2, 2004, pp. 1980–1987, doi: 10.1109/CEC.2004.1331139.

[26] Derrac, J.—García, S.—Molina, D.—Herrera, F.: A Practical Tutorial on the Use of Non-Parametric Statistical Tests as a Methodology for Comparing Evolu-

tionary and Swarm Intelligence Algorithms. Swarm and Evolutionary Computation, Vol. 1, 2011, No. 1, pp. 3–18, doi: 10.1016/j.swevo.2011.02.002.

[27] Liu, B.—Wang, L.—Jin, Y.-H.—Tang, F.—Huang, D.-X.: Improved Particle Swarm Optimization Combined with Chaos. Chaos, Solitons and Fractals, Vol. 25, 2005, No. 5, pp. 1261–1271, doi: 10.1016/j.chaos.2004.11.095.

[28] Erol, O.—Eksin, I.: A New Optimization Method: Big Bang – Big Crunch. Advances in Engineering Software, Vol. 37, 2006, No. 2, pp. 106–111, doi: 10.1016/j.advengsoft.2005.04.005.

[29] Wang, G.-G.—Guo, L.—Gandomi, A. H.—Hao, G.-S.—Wang, H.: Chaotic Krill Herd Algorithm. Information Sciences, Vol. 274, 2014, pp. 17–34, doi: 10.1016/j.ins.2014.02.123.

[30] Gao, S.—Vairappan, C.—Wang, Y.—Cao, Q.—Tang, Z.: Gravitational Search Algorithm Combined with Chaos for Unconstrained Numerical Optimization. Applied Mathematics and Computation, Vol. 231, 2014, pp. 48–62, doi: 10.1016/j.amc.2013.12.175.

[31] Alatas, B.: Chaotic Bee Colony Algorithms for Global Numerical Optimization. Expert Systems with Applications, Vol. 37, 2010, No. 8, 2010, pp. 5682–5687, doi: 10.1016/j.eswa.2010.02.042.

[32] Saremi, S.—Mirjalili, S.—Lewis, A.: Biogeography-Based Optimisation with Chaos. Neural Computing and Applications, Vol. 25, 2014, pp. 1077–1097, doi: 10.1007/s00521-014-1597-x.

[33] Talatahari, S.—Kaveh, A.—Sheikholeslami, R.: Chaotic Imperialist Competitive Algorithm for Optimum Design of Truss Structures. Structural and Multidisciplinary Optimization, Vol. 46, 2012, No. 3, pp. 355–367, doi: 10.1007/s00158-011-0754-4.

[34] Yu, H.—Yu, Y.—Liu, Y.—Wang, Y.—Gao, S.: Chaotic Grey Wolf Optimization. Proceedings of the 2016 IEEE International Conference on Progress in Informatics and Computing (PIC 2016), 2017, pp. 108–113, doi: 10.1109/PIC.2016.7949476.

[35] Storn, R.—Price, K.: Differential Evolution – A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces. Journal of Global Optimization, Vol. 11, 1997, pp. 341–359, doi: 10.1023/A:1008202821328.

[36] Jia, D.—Zheng, G.—Khan, M. K.: An Effective Memetic Differential Evolution Algorithm Based on Chaotic Local Search. Information Sciences, Vol. 181, 2011, No. 15, pp. 3175–3187, doi: 10.1016/j.ins.2011.03.018.

[37] Dorigo, M.—Birattari, M.—Stützle, T.: Ant Colony Optimization. IEEE Computational Intelligence Magazine, Vol. 1, 2006, No. 4, pp. 28–39, doi: 10.1109/MCI.2006.329691.

[38] Cai, J.—Ma, X.—Li, L.—Yang, Y.—Peng, H.—Wang, X.: Chaotic Ant Swarm Optimization to Economic Dispatch. Electric Power Systems Research, Vol. 77, 2007, No. 10, pp. 1373–1380, doi: 10.1016/j.epsr.2006.10.006.

[39] Karaboga, D.—Basturk, B.: A Powerful and Efficient Algorithm for Numerical Function Optimization: Artificial Bee Colony (ABC) Algorithm. Journal of Global Optimization, Vol. 39, 2007, No. 3, pp. 459–471, doi: 10.1007/s10898-007-9149-x.

[40] Atashpaz-Gargari, E.—Lucas, C.: Imperialist Competitive Algorithm: An Algorithm for Optimization Inspired by Imperialistic Competition. 2007 IEEE Congress on Evolutionary Computation (CEC 2007), 2007, pp. 4661–4667, doi: 10.1109/CEC.2007.4425083.

[41] Talatahari, S.—Farahmand Azar, B.—Sheikholeslami, R.—Gandomi, A. H.: Imperialist Competitive Algorithm Combined with Chaos for Global Optimization. Communications in Nonlinear Science and Numerical Simulation, Vol. 17, 2012, No. 3, pp. 1312–1319, doi: 10.1016/j.cnsns.2011.08.021.

[42] Simon, D.: Biogeography-Based Optimization. IEEE Transactions on Evolutionary Computation, Vol. 12, 2008, No. 6, pp. 702–713, doi: 10.1109/TEVC.2008.919004.

[43] Rashedi, E.—Nezamabadi-pour, H.—Saryazdi, S.: GSA: A Gravitational Search Algorithm. Information Sciences, Vol. 179, No. 13, 2009, pp. 2232–2248, doi: 10.1016/j.ins.2009.03.004.

[44] Yang, X.-S.: A New Meta-Heuristic Bat-Inspired Algorithm. In: González, J. R., Pelta, D. A., Cruz, C., Terrazas, G., Krasnogor, N. (Eds.): Nature Inspired Cooperative Strategies for Optimization (NICSO 2010). Springer, Berlin, Heidelberg, Studies in Computational Intelligence, Vol. 284, 2010, pp. 65–74, doi: 10.1007/978-3-642-12538-6_6.

[45] Rezaee Jordehi, A.: Chaotic Bat Swarm Optimisation (CBSO). Applied Soft Computing, Vol. 26, 2015, pp. 523–530, doi: 10.1016/j.asoc.2014.10.010.

[46] Yang, X.-S.—Deb, S.: Engineering Optimisation by Cuckoo Search. International Journal of Mathematical Modelling and Numerical Optimisation (IJMMNO), Vol. 1, 2010, No. 4, pp. 330–343, doi: 10.1504/IJMMNO.2010.035430.

[47] Wang, G.-G.—Deb, S.—Gandomi, A. H.—Zhang, Z.—Alavi, A. H.: Chaotic Cuckoo Search. Soft Computing, Vol. 20, 2016, pp. 3349–3362, doi: 10.1007/s00500-015-1726-1.

[48] Yang, X.-S.: Firefly Algorithm, Stochastic Test Functions and Design Optimisation. International Journal of Bio-Inspired Computation, Vol. 2, 2010, No. 2, pp. 78–84, doi: 10.1504/IJBIC.2010.032124.

[49] Gandomi, A. H.—Yang, X.-S.—Talatahari, S.—Alavi, A. H.: Firefly Algorithm with Chaos. Communications in Nonlinear Science and Numerical Simulation, Vol. 18, 2013, No. 1, pp. 89–98, doi: 10.1016/j.cnsns.2012.06.009.

[50] Thangaraj, R.—Pant, M.—Abraham, A.—Bouvry, P.: Particle Swarm Optimization: Hybridization Perspectives and Experimental Illustrations. Applied Mathematics and Computation, Vol. 217, 2011, No. 12, pp. 5208–5226, doi: 10.1016/j.amc.2010.12.053.

[51] Alatas, B.—Akin, E.—Ozer, A. B.: An Adaptive Chaos Embedded Particle Swarm Optimization Algorithm. Chaos, Solitons and Fractals, Vol. 40, 2009, No. 4, pp. 1715–1734, doi: 10.1016/j.chaos.2007.09.063.

[52] Gandomi, A. H.—Alavi, A. H.: Krill Herd: A New Bio-Inspired Optimization Algorithm. Communications in Nonlinear Science and Numerical Simulation, Vol. 17, 2012, No. 12, pp. 4831–4845, doi: 10.1016/j.cnsns.2012.05.010.

[53] Mirjalili, S.—Mirjalili, S. M.—Lewis, A.: Grey Wolf Optimizer. Advances in Engineering Software, Vol. 69, 2014, pp. 46–61, doi: 10.1016/j.advengsoft.2013.12.007.

[54] Kohli, M.—Arora, S.: Chaotic Grey Wolf Optimization Algorithm for Constrained Optimization Problems. Journal of Computational Design and Engineering, Vol. 5, 2018, No. 4, pp. 458–472, doi: 10.1016/j.jcde.2017.02.005.

[55] Mirjalili, S.—Gandomi, A. H.—Mirjalili, S. Z.—Saremi, S.—Faris, H.—Mirjalili, S. M.: Salp Swarm Algorithm: A Bio-Inspired Optimizer for Engineering Design Problems. Advances in Engineering Software, Vol. 114, 2017, pp. 163–191, doi: 10.1016/j.advengsoft.2017.07.002.

[56] Sayed, G. I.—Khoriba, G.—Haggag, M. H.: A Novel Chaotic Salp Swarm Algorithm for Global Optimization and Feature Selection. Applied Intelligence, Vol. 48, 2018, No. 10, pp. 3462–3481, doi: 10.1007/s10489-018-1158-6.

[57] Yaghoobi, S.—Mojallali, H.: Tuning of a PID Controller Using Improved Chaotic Krill Herd Algorithm. Optik, Vol. 127, 2016, No. 11, pp. 4803–4807, doi: 10.1016/j.ijleo.2016.01.055.

[58] Suganthan, P. N.—Hansen, N.—Liang, J. J.— Deb, K.—Chen, Y. P.—Auger, A.—Tiwari, S.: Problem Definitions and Evaluation Criteria for the CEC 2005 Special Session on Real-Parameter Optimization. Technical Report. Nanyang Technological University, Singapore, 2005, `http://www.ntu.edu.sg/home/EPNSugan`.

[59] Chen, Q.—Liu, B.—Zhang, Q.—Liang, J.: Evaluation Criteria for CEC 2015 Special Session and Competition on Bound Constrained Single-Objective Computationally Expensive Numerical Optimization. Singapore, 2015. `http://web.mysites.ntu.edu.sg/epnsugan/PublicSite/SharedDocuments/CEC-2015/ExpensiveProblems/CEC15-singleobjectiveoptimizationcompetition-expensive.pdf`.

[60] Igel, C.—Hansen, N.—Roth, S.: Covariance Matrix Adaptation for Multi-Objective Optimization. Evolutionary Computation, Vol. 15, 2007, No. 1, pp. 1–28, doi: 10.1162/evco.2007.15.1.1.

[61] Qin, A. K.—Suganthan, P. N.: Self-Adaptive Differential Evolution Algorithm for Numerical Optimization. 2005 IEEE Congress on Evolutionary Computation, Vol. 2, 2005, pp. 1785–1791, doi: 10.1109/CEC.2005.1554904.

[62] Brest, J.—Greiner, J.—Boskovic, B.—Mernik, M.—Zumer, V.: Self-Adapting Control Parameters in Differential Evolution: A Comparative Study on Numerical Benchmark Problems. IEEE Transactions on Evolutionary Computation, Vol. 10, 2006, No. 6, pp. 646–657, doi: 10.1109/TEVC.2006.872133.

[63] Kumar, M.—Kulkarni, A. J.—Satapathy, S. C.: Socio Evolution and Learning Optimization Algorithm: A Socio-Inspired Optimization Methodology. Future Generation Computer Systems, Vol. 81, 2018, pp. 252–272, doi: 10.1016/j.future.2017.10.052.

[64] Liang, J. J.—Qu, B. Y.—Suganthan, P. N.—Hernández-Díaz, A. G.: Problem Definitions and Evaluation Criteria for the CEC 2013 Special Session and Competition on Real-Parameter Optimization. Technical Report 201212, Computational

Intelligence Laboratory, Zhengzhou University, Zhengzhou, China and Nanyang Technological University, Singapore, 2013.

Hojjat Emami received his B.Sc. degree in software engineering and his M.Sc. degree in artificial intelligence, both from the University of Tabriz, Iran. He received the Ph.D. degree in artificial intelligence from the Malek-Ashtar University of Technology, Tehran, Iran. His research interests are multi-agent systems, data mining, machine learning and optimization.