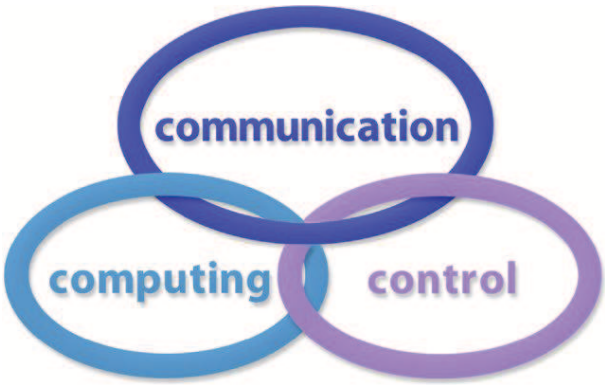
INTERNATIONAL JOURNAL

of

COMPUTERS COMMUNICATIONS & CONTROL

ISSN 1841-9836, e-ISSN 1841-9844



A Bimonthly Journal
With Emphasis on the Integration of Three Technologies

Year: 2019 Volume: 14 Issue: 2 Month: April

This journal is a member of, and subscribes to the principles of, the Committee on Publication Ethics (COPE).

http://univagora.ro/jour/index.php/ijccc/

CCC Publications

BRIEF DESCRIPTION OF JOURNAL

Publication Name: International Journal of Computers Communications & Control.
Acronym: IJCCC; **Starting year of IJCCC:** 2006.
ISO: Int. J. Comput. Commun. Control; **JCR Abbrev:** INT J COMPUT COMMUN.
International Standard Serial Number: ISSN 1841-9836, e-ISSN 1841-9844.
Publisher: CCC Publications - Agora University of Oradea.
Publication frequency: Bimonthly: Issue 1 (February); Issue 2 (April); Issue 3 (June); Issue 4 (August); Issue 5 (October); Issue 6 (December).
Founders of IJCCC: Ioan DZITAC, Florin Gheorghe FILIP and Misu-Jan MANOLESCU.

Indexing/Coverage:

- Since 2006, Vol. 1 (S), IJCCC is covered by Clarivate Analytics and is indexed in ISI Web of Science/Knowledge: Science Citation Index Expanded.

  2018 Journal Citation Reports® Science Edition (Clarivate Analytics, 2017):
  Subject Category: (1) Automation & Control Systems: Q4(2009, 2011, 2012, 2013, 2014, 2015), **Q3(2010, 2016, 2017)**; (2) Computer Science, Information Systems: Q4(2009, 2010, 2011, 2012, 2015), **Q3(2013, 2014, 2016, 2017)**.

  Impact Factor/3 years in JCR: 0.373(2009), 0.650 (2010), 0.438(2011); 0.441(2012), 0.694(2013), 0.746(2014), 0.627(2015), 1.374(2016), **1.29 (2017)**.
  Impact Factor/5 years in JCR: 0.436(2012), 0.622(2013), 0.739(2014), 0.635(2015), 1.193(2016), **1.179(2017)**.

- Since 2008 IJCCC is indexed by Scopus: **CiteScore 2017 = 1.04**.
  Subject Category:
  (1) Computational Theory and Mathematics: Q4(2009, 2010, 2012, 2015), **Q3(2011, 2013, 2014, 2016, 2017)**;
  (2) Computer Networks and Communications: Q4(2009), Q3(2010, 2012, 2013, 2015), **Q2(2011, 2014, 2016, 2017)**;
  (3) Computer Science Applications: Q4(2009), **Q3(2010, 2011, 2012, 2013, 2014, 2015, 2016, 2017)**.

  SJR: 0.178(2009), 0.339(2010), 0.369(2011), 0.292(2012), 0.378(2013), 0.420(2014), 0.263(2015), 0.319(2016), 0.326 (2017).

- Since 2007, 2(1), IJCCC is indexed in EBSCO.

Focus & **Scope:** International Journal of Computers Communications & Control is directed to the international communities of scientific researchers in computers, communications and control, from the universities, research units and industry. To differentiate from other similar journals, the editorial policy of IJCCC encourages the submission of original scientific papers that focus on the integration of the 3 "C" (Computing, Communications, Control).

In particular, the following topics are expected to be addressed by authors:

(1) Integrated solutions in computer-based control and communications;

(2) Computational intelligence methods & Soft computing (with particular emphasis on fuzzy logic-based methods, computing with words, ANN, evolutionary computing, collective/swarm intelligence, membrane computing, quantum computing);

(3) Advanced decision support systems (with particular emphasis on the usage of combined solvers and/or web technologies).

EDITORIAL ADDRESS:
Agora University, Cercetare Dezvoltare Agora, Tineretului 8, 410526 Oradea, Bihor, Romania,
Tel./ Fax: +40 359101032, E-mail: ijccc@univagora.ro, rd.agora@univagora.ro
URL: http://univagora.ro/jour/index.php/ijccc/

EDITORIAL BOARD OF IJCCC (MEMBERS, 2018):

Constantin GAINDRIC
IMMAS, Republic of MOLDOVA
Kishinev, 277028, Academiei 5
gaindric@math.md

Xiao-Shan GAO
Academia Sinica, CHINA
Beijing 100080, China
xgao@mmrc.iss.ac.cn

Enrique HERRERA-VIEDMA
University of Granada, SPAIN
Av. del Hospicio, s/n, 18010 Granada
viedma@decsai.ugr.es

Kaoru HIROTA
Tokyo Institute of Tech., JAPAN
G3-49,4259 Nagatsuta
hirota@hrt.dis.titech.ac.jp

Arturas KAKLAUSKAS
VGTU, LITHUANIA
Sauletekio al. 11, LT-10223 Vilnius
arturas.kaklauskas@vgtu.lt

Gang KOU
SWUFE, CHINA
Chengdu, 611130
kougang@swufe.edu.cn

Heeseok LEE
KAIST, SOUTH KOREA
85 Hoegiro, Seoul 02455
hsl@business.kaist.ac.kr

George METAKIDES
University of Patras, GREECE
Patra 265 04, Greece
george@metakides.net

Shimon Y. NOF
Purdue University, USA
610 Purdue Mall, West Lafayette
nof@purdue.edu

Stephan OLARIU
Old Dominion University, USA
Norfolk, VA 23529-0162
olariu@cs.odu.edu

Gheorghe PĂUN
Romanian Academy, ROMANIA
IMAR, Bucharest, PO Box 1-764
gpaun@us.es

Mario de J. PEREZ JIMENEZ
University of Seville, SPAIN
Avda. Reina Mercedes s/n, 41012
marper@us.es

Radu-Emil PRECUP
Pol. Univ. of Timisoara, ROMANIA
Bd. V. Parvan 2, 300223
radu.precup@aut.upt.ro

Radu POPESCU-ZELETIN
Technical University Berlin, GERMANY
Fraunhofer Institute for Open CS
rpz@cs.tu-berlin.de

Imre J. RUDAS
Obuda University, HUNGARY
Budapest, Becsi ut 96b, 1034
rudas@bmf.hu

Yong SHI
Chinese Academy of Sciences, CHINA
Beijing 100190
yshi@gucas.ac.cn, yshi@unomaha.edu

Bogdana STANOJEVIC
Serbian Academy of SA, SERBIA
Kneza Mihaila 36, Beograd 11001
bgdnpop@mi.sanu.ac.rs

Athanasios D. STYLIADIS
University of Kavala, GREECE
65404 Kavala
styliadis@teikav.edu.gr

Gheorghe TECUCI
George Mason University, USA
University Drive 4440, Fairfax VA
tecuci@gmu.edu

Horia-Nicolai TEODORESCU
Romanian Academy, ROMANIA
Iasi Branch, Bd. Carol I 11, 700506
hteodor@etc.tuiasi.ro

Dan TUFIŞ
Romanian Academy, ROMANIA
13 Septembrie, 13, 050711 Bucharest
tufis@racai.ro

Edmundas K. ZAVADSKAS
VGTU, LITHUANIA
Sauletekio ave. 11, LT-10223 Vilnius
edmundas.zavadskas@vgtu.lt

INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL
ISSN 1841-9836, e-ISSN 1841-9844, Volume 14, Issue 2, April, 2019.

Contents

INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL
ISSN 1841-9836, e-ISSN 1841-9844, 14(2), 141-153, April 2019.

Gene Sequences Parallel Alignment Model Based on Multiple Inputs and Outputs

X.L. Feng, J. Gao

Xiaolong Feng, Jing Gao*
College of Computer and Information Engineering
Inner Mongolia Agricultural University
Hohhot 010018, China
*Corresponding author: gaojing@imau.edu.cn

Abstract: Bioinformatics computing is a kind of big data processing problem, which usually has the characteristics of large data scale, large computational load and long computational time. Therefore, the use of big data technology in bioinformatics computing has gradually become a research hotspot, and using Hadoop for gene sequence alignment is one of it. It is a common way to use various tools to complete a job in the field of Biocomputing. In most studies of parallel alignment of gene sequences using Hadoop, third-party tools are also needed. However, there are few methods using Hadoop independently to complete gene sequences alignment. Adding data processing with other tools to Hadoop workflow not only affects the improvement of computing performance, but also complicates the application. In this paper, a parallel alignment model of gene sequences based on multiple inputs and outputs is proposed, which can independently complete parallel alignment of gene sequences in Hadoop platform without using other tools. This model not only simplifies the process flow of gene sequence alignment, but also improves the performance compared with other methods. This paper describes in detail the method of manipulating gene sequences with multiple inputs and outputs modes on Hadoop platform and the design of a computing model based on this method, and proves the superiority of this model through experiments.
Keywords: Multiple inputs and outputs, MapReduce, gene sequence alignment, short reads mapping, BWA (Burrows-Wheeler aligner), parallel computing.

1 Introduction

Gene sequence alignment is a time-consuming task in gene sequence analysis. With the rapid development of gene sequencing technology in terms of capacity and speed, non-parallel computing method has become a bottleneck in the flow of bioinformatics analysis. It is an urgent need to design and develop a set of stable, efficient and scalable calculation methods to solve this problem. Hadoop Distributed Parallel Computing Framework is a solution to this problem, because it provides a general method for processing large-scale data [17]. It can greatly improve the performance of large-scale data computing such as gene sequences and improve the scalability of computing methods [16]. Hadoop is a distributed computing infrastructure released by Apache Foundation. It is a high fault-tolerant and high throughput open source computing framework deployable on low-cost hardware platforms. It is very suitable for storage and computation of applications with large data sets. It provides HDFS distributed file system, YARN resource scheduling manager and MapReduce computing model [8].

BWA (Burrows-Wheeler aligner) algorithm is a gene sequence alignment algorithm widely used in bioinformatics analysis. It can map a large number of short reads gene sequences to large-scale genomes [11]. At present, mature BWA tools are single-machine serial execution or multi-threaded parallel execution, and scalable distributed parallel computing methods are being

studied. Like many other bioinformatics computations, these BWA parallel computational methods require multiple tools to perform a data analysis task together. Data needs to be processed by tools before input and after output. This not only makes the calculation process cumbersome, but also affects the efficiency of data analysis. Therefore, a parallel computing model running on Hadoop framework is proposed in this paper, which can independently complete the job of gene sequence alignment. This model can be used in distributed parallel computing of BWA algorithm. In this model, the method of data multiple inputs and outputs is used, which meets the requirement of gene data operation without using other tools. Meanwhile, a MapReduce computing model matching this data input and output mode is designed, which improves the parallel computing performance of BWA algorithm. The design of this BWA parallel computing model considers the following three requirements. Firstly, the model is superior to BWA algorithm and other BWA-based parallel computing methods in performance and scalability, including BWA's own single-machine multi-threaded parallel computing method. Secondly, the results of the model should be compatible with the traditional bioinformatics analysis process. Because different versions of BWA tool contain different algorithm and characteristics, and different analysis work depends on different algorithms, so the model should provide a configurable interface to invoke the desired algorithm, instead of encapsulating only a particular algorithm. Thirdly, the model should ensure the integrity of functions. Avoid using other tools and intervention in the calculation process, and complete all tasks independently by the model. This allows users to concentrate on the scientific issues without considering the use and compatibility of multiple tools. The evaluation of this model should be compared with BWA algorithm and other parallel computing methods based on BWA algorithm in terms of performance and scalability. The advantages of the computing model are illustrated by time-consuming, speedup ratio and parallel efficiency.

2 Research background

BWA is a commonly used gene sequence alignment tool in bioinformatics analysis. It contains three algorithms: BWA-backtrack, BWA-SW and BWA-MEM. The first algorithm is suitable for sequence mapping whose length is less than 100 bp, and the latter two are suitable for sequence mapping whose length is longer. BWA-MEM is more efficient than other algorithms in sequence processing over 100 bp. The BWA mapping is a time-consuming step in gene sequence analysis, improve the efficiency of mapping has become the key to improve the biological information analysis. For this reason, BWA software also provides multithreaded parallel computing method. But this method is limited to the capacity of single-machine, and does not support distributed expansion. Therefore, its performance is not very high, especially for large genomes, which takes a long time for alignment jobs, and it may also fail because of single point failure.

Hadoop is a suitable platform for bioinformatics computing in terms of data scale, job characteristics and cost of implementation. MapReduce is a programming model suitable for handling large amounts of semi-structured data sets. The functional programming method of MapReduce is a simple way for developer. Users can implement parallel execution of computing tasks by writing the Map and Reduce functions. It provides an abstract parallel programming interface for operation, and implements the computation and processing of large-scale data in a simple way. In Map and Reduce functions, users can freely and flexibly add in parallel operation of data, which facilitates the processing of semi-structured data.

The input of BWA algorithm is usually a sequence file in FASTQ format. It is the result file of gene sequencing. The sequencing results may be single-end sequence or pair-end sequence for different sequencing methods [5, 13]. The output of BWA algorithm is SAM format file, which mainly records the location and hit times of short reads sequence mapped to reference sequence.

As to the data format, the input and output data of BWA algorithm are semi-structured data, which is suitable for MapReduce programming model. However, in the current BWA parallel computing research, there is no solution to the problem of data unified processing. Data must be processed by other tools to adapt to MapReduce programming model. If we can design a MapReduce data input and output method suitable for gene sequence processing, it will be very convenient for users to use Hadoop for bioinformatics analysis.

3 Literature review

In the research of using Hadoop to improve BWA algorithm, there are three typical representatives: BigBWA [1], Halvade [9] and SEAL [15]. BigBWA uses JNI interface to invoke BWA source code to implement distributed parallel computing of sequence alignment, which significantly improves the performance of BWA algorithm. The disadvantage is that you need to use tools to change the original format of input data before computing starts. SEAL implements MapReduce model of BWA with Python. The disadvantage is that it cannot satisfy all Hadoop native interfaces, and its running efficiency is lower than that of Java or $C++$ applications. This model only encapsulates a specific version of BWA software and cannot support the application of new versions, such as BWA-MEM for long sequence alignment. Halvade is a Hadoop-based gene sequence alignment framework developed with Java. It performs data split and sequence alignment in Map function, calls different gene analysis programs in Reduce function, and has many functions. But in data input and distribution, in order not to be restricted by Hadoop's specifications, it designed a platform-independent program "Halvade Uploader" to complete data distribution. Although multi-threading is adopted, it cannot support distributed extension and is not consistent with the Hadoop platform.

The common feature of these studies is that they all use Hadoop platform to parallelize computing tasks, which greatly improves the performance of BWA algorithm compared with serial execution. However, in these studies, third-party programs or applications independent of computing platforms are used for data preprocessing or post-processing, which makes the computing model not uniform as a whole, and also affects the improvement of computing performance to a certain extent. The reason for this approach is that Hadoop does not provide a way to directly process gene sequences, while third-party tools can easily cope with it. Taking BigBWA as an example, input data need to be preprocessed using Python program. It makes the single-end gene sequence form a single-line structure linked by $< sep >$ markers, such as $line1 < sep > line2 < sep > line3 < sep > line4$, and pair-end gene sequence form a single-line structure linked by $< part >$ markers, such as $left-end-of-sequence1 < part > right-end-of-sequence1$. At the same time, two data files of pair-end sequence are merged into one data file. The reason is that single file and single-line structure are the most convenient way for Hadoop to read directly. These additional tags need to be removed when they enter the MapReduce computing model to be accepted by the alignment algorithm. This increases the overhead of format processing in computing model. Like input data, BigBWA's output data also needs to be processed using Python programs. It merges SAM files on multiple nodes into one result file. The parallel computing model designed in this paper completes the process of data input, data distribution, distributed computing and result processing of gene sequence alignment task only with Hadoop API. In this model, input data need not be pre-processed, and can be directly input by FASTQ format single-end or pair-end sequence files. The overhead of format processing is also reduced in MapReduce computing model. Output data merging is also done without tools. Computing tasks are automatically executed without interference in the process.

4 Problem descriptions

The task of short sequence alignment is to map a large number of single-end or pair-end short reads sequences to the reference genome, and then carries out subsequent biological significance analysis. The input of alignment algorithm is short reads sequence file and reference sequence file, and the output is mapping result file. The traditional serial alignment algorithm usually takes a long time because of the huge amount of data. The principle of short sequence alignment is to find the exact position of each short reads sequence in the reference sequence by global alignment of each short reads sequence with the reference genome [12]. The principle of gene sequence alignment based on Hadoop is to distribute short reads sequence data to multiple nodes of distributed cluster, then map to reference sequence independently on each node and form their own result files. Finally, the result files on each node are aggregated to form a result file [2]. Short reads sequence is FASTQ format file. The pair-end sequence consists of left-end and right-end files. Each file consists of many short reads sequences, each of which has a fixed structure. A short reads sequence file can be regarded as a set of m sequences, and the left-end sequence can be described as $L\{l_1, l_2, l_3, ..., l_m\}$, the right-end sequence can be described as $R\{r_1, r_2, r_3, ..., r_m\}$. Then the pair-end sequence is described as $READS\{L, R\}$. Single-end sequence can be regarded as a special case with only left-end. Reference sequence is a long sequence that stores complete genetic information, REF represents reference sequence, ALN represents alignment algorithm, S represents the set of alignment results $\{s_1, s_2, s_3, ..., s_m\}$. Then the alignment problem discussed is described as

$$s_i = ALN(l_i, r_i, REF), i = 1, 2, 3...m$$

In order to distribute parallel execution of alignment tasks, $READS$ set can be divided into n subsets $D\{d_1, d_2, d_3, ..., d_n\}$ and distributed to n work nodes in the cluster. Each node will be allocated to $k = m/n$ short reads sequences if equal division of dataset is adopted. Then the data set on the work node i can be represented as

$$d_i = \{L_i, R_i\}, i = 1, 2, 3...n$$

$$L_i = \{l_{(i-1)*k+1}, L_{(i-1)*k+2}, ..., l_{i*k}\}, k = m/n$$

$$R_i = \{r_{(i-1)*k+1}, r_{(i-1)k+2}, ..., r_{i*k}\}, k = m/n$$

Computing task on node i is represented as $s_i = ALN(d_i, REF)$. Computing tasks on all nodes are executable in parallel. When the task is completed, the s_i on each node can be merged into a result file, which is compatible with the traditional biological analysis work.

To implement the distributed parallelization of alignment algorithm, four main problems need to be solved:

(1) How to input the gene sequence file directly without preprocessing?

(2) How to distribute the sequence in READS set to the work nodes?

(3) How to execute the alignment algorithm?

(4) How to merge the alignment results on the work nodes?

In data input, a single-end sequence can be regarded as a special case of a pair-end sequence. So the main problem is how to input two or more files with Hadoop APIs without preprocessing. The change of data input method will lead to the change of subsequent data calculation method, so it needs to be considered comprehensively. Hadoop provides APIs for multiple data sources to read at the same time, which can solve this problem. It also needs to design programs to meet the requirements of gene sequence operation. Gene sequences from different data sources need to be treated differently in the algorithm. It is necessary to ensure that the left-end sequence and the right-end sequence of a gene sequence can be recognized and integrated.

If there are m short reads sequences and n work nodes in an alignment job, the task of data distribution is to distribute m sequences to n work nodes. In order to get higher efficiency, the data distribution requires minimizing the amount of data movement and keep load balance on the work nodes. The simple way is to distribute m sequences in full to all work nodes. The disadvantage of this method is that it takes up a large amount of disk space, has large network traffic and takes a long time. The advantage of this method is that it does not need to design a distribution algorithm, and does not need to move data in the process of task execution. The ideal method is to distribute the necessary sequence to the designated nodes, so as to avoid moving data during task scheduling. Custom partition in Hadoop, which allows data to be distributed to designate nodes according to computing requirements, can solve the problem of data distribution very well.

In the phase of executing alignment algorithm, in order to ensure the compatibility of alignment results and the integrity of alignment function, the best way is to call the existing traditional alignment program without changing the source code. Invoking alignment program on distributed cluster can be implemented by JNI, PIPLE or SHELL. The collection and merging of result files can be accomplished on HDFS by Hadoop file manipulation.

5 Method

The goal of model design is to design a stable, reliable, efficient and scalable distributed computing model, so that short sequence alignment algorithm can be distributed and parallel implemented on Hadoop platform. The model does not need third-party tools to automate processes, including data input, data distribution, distributed computing and result collection.

The following assumptions are made for the configuration or characteristics of distributed cluster:

– Computing framework runs in distributed cluster with one name node and several data nodes with the same capacity.

– Each alignment operation can be independent of other tasks.

– The reference database is pre-deployed to the system, and all alignment tasks can be performed by any data node.

– Short reads sequence files can be split into multiple sequences and reassembled.

The function of Hadoop platform provides great convenience for designing distributed parallel model of alignment algorithm. Considering the characteristics of platform and alignment job, a computing model is designed as shown in Figure 1. HDFS is used to store input data and results, which is convenient for data distribution and sharing in Hadoop platform. Before the job submission, the pre-deployment work should be completed, that is, the reference sequence and alignment software should be deployed to each working node in advance, and the short reads sequence file should be uploaded to HDFS to facilitate the distributed deployment. In data input, Hadoop multi-input API is used to read multiple sequence files directly without data preprocessing. Two or more data files can be stored in HDFS for multiple inputs as shown in Figure 1, L and R represent two files of pair-end sequence respectively. Single-end sequence input is considered as a special case of multiple inputs. In Mapper, left-end or right-end tags are added for key-value pairs. Then the data is partitioned according to the key of the pair by using the custom partitioning algorithm, and the partitioned data is distributed to the work nodes of the cluster. In Reducer, multi-output is used to transfer the partition data to the alignment algorithm, and the alignment algorithm is invoked on the working node to implement the distributed computing of each partition. Finally, the results on each node are collected into HDFS and merged in one file.

The design of computing model needs to take advantage of the functions provided by the platform and comply with its programming specifications [20]. Therefore, according to the model shown in Figure 1, a processing flow is designed, which includes seven steps: pre-deployment, data input, Mapper processing, data partition, Merge processing, Reducer processing, and result processing [14], as shown in Figure 2.

Two classes, $FileInputFormat$ and $MultipleInputs$, are provided in Hadoop API to support multiple inputs. The former uses unified Mapper processing, while the latter supports independent Mapper processing. In this paper, the $addInputPath$ method of $FileInputFormat$ class is used to implement multi-input. By organizing the paths of multiple data files into an array, and then passing the array as a parameter to the function, the input class can read data from multiple data sources. The input gene sequence whether left or right, forms a key-value pair with offset as key and sequence content as value.

In Mapper processing, pair-end sequences are identified as left-end or right-end and labeled, while single-end sequences are not labeled, such as Algorithm 1. The tags added in Mapper is the basis for subsequent implementation of data partition constraints and data multiple outputs. After Mapper processing, the left-end and the right-end of a pair-end sequences form key-value pairs with the same key. Because a pair-end sequence has the same offset in two files.

Algorithm 1 The Map algorithm

```
INPUT: (key, value)
OUTPUT: (key, value')
if value stores a pair-end sequence then
    if value stores a left-end sequence then
        value' = addTag(value, left-end-tag)
    else
        value' = addTag(value, right-end-tag)
    end if
else
    value' = value
end if
Context.write(key, value')
```

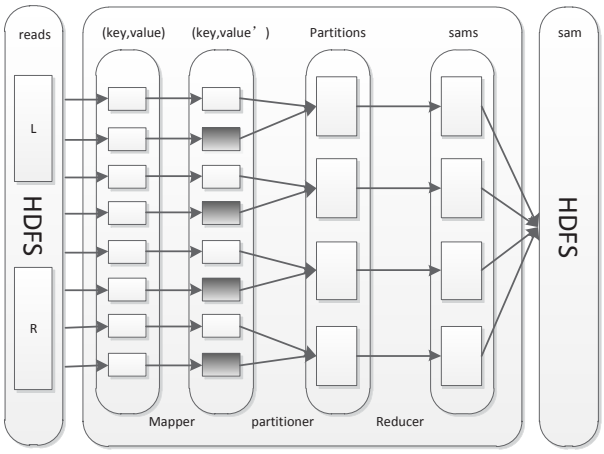


Figure 1: MapReduce model for short reads gene sequence alignment

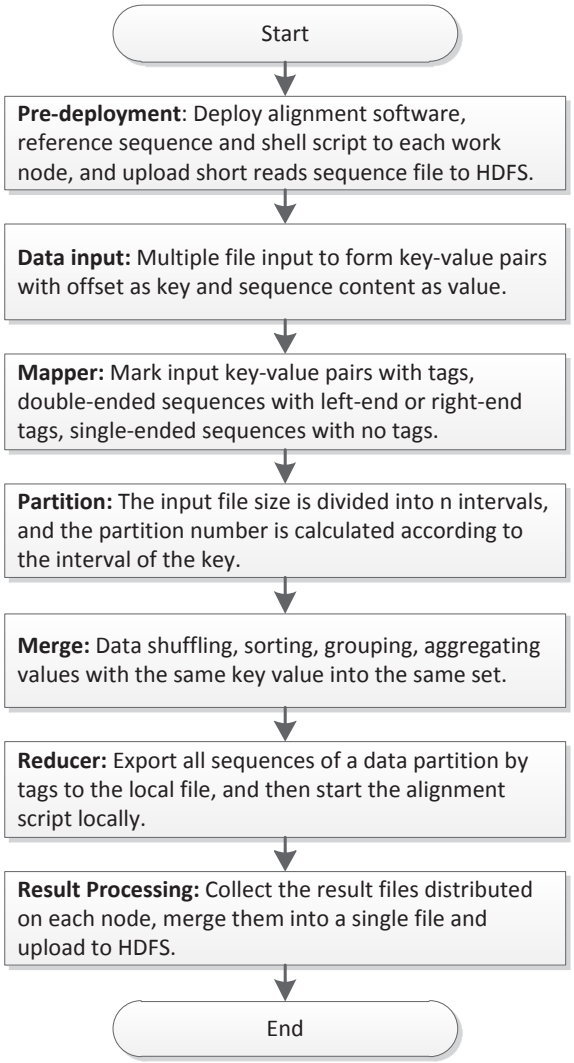


Figure 2: Flow chart of short reads gene sequence parallel alignment

Partitioner is a means of data distribution provided by Hadoop platform [10]. The partition classes built in the platform, such as $HashPartitioner$ and $BinaryPartitioner$, are not suitable for the distribution of pair-end sequences. The operation of gene sequences has some constraints on data partitioning, which is not satisfied by Hadoop's partition classes. It is necessary to customize partition classes according to the requirements of gene sequence operation.

The constraints of data partitioning include:

- The number of left-end sequences in each partition is the same as that of right-end sequences;

- The position of sequences in a partition remains unchanged relative to that in sequence file;

- The left-end and the right-end of a pair-end sequence must be allocated to the same partition.

Since it has been assumed that the capacity of each work node in the cluster is the same, data is divided into equal partitions. The partitioning algorithm is shown in Algorithm 2. Firstly, the intervals of offsets are calculated by the size of sequence file and the number of partitions. Then partition is calculated with key in $(key, value')$. Key represents the position of a sequence in the sequence file. The partition number can be obtained by judging the offset interval of key. Sequences at the same location of the pair-end sequence have the same key, so they are assigned to the same partition.

Algorithm 2 The partition algorithm

```
INPUT: (key, value')
OUTPUT: partitionNum
LET fileSize ← The size of input file
LET partitionNum ← 0
LET n ← The number of nodes
while partitonNum < n do
    if key < (partitionNum + 1)fileSize/n then
        return partitionNum
    end if
    partitionNum++
end while
return n − 1
```

The Merge phase will shuffle, sort, and group the data in the partition based on the key. The result of processing is that sequences with the same key are aggregated into the same set. For a pair-end sequence, a key corresponds to a set containing its left-end and right-end sequence, and the key-value pair is in the form of $(key, value'[])$.Moreover, all sequences are ordered in the partition, which ensures that the relative positions of the sequences in the partition remain unchanged. Input in reducer is a data partition and aggregated by key. If the input is a single-end sequence, the sequence in the partition will be written to the local file one by one, and if the input is a pair-end sequence, it will be identified by tags added in Mapper and output to different local files in multi-output method, as shown in Algorithm 3. Therefore, the output of Reduce function to a single-end sequence is a single local file, the content is the data partition on the node, and the output to a pair-end sequence is two local files.

After the local data file is generated, the sequence alignment task is started in the cleanup function of Reducer, and the task is executed by shell call. Shell script can be modified at any time according to the requirement of software version or parameter configuration, which makes

Algorithm 3 The reduce algorithm

```
INPUT: (key, value'[])
OUTPUT: left − part_i, right − part_i OR part_i
for all value in value'[] do
    if value stores a pair-end sequence then
        if value stores a left-end sequence then
            MutiOutput.write("left-part",value)
        else
            MutiOutput.write("right-part",value)
        end if
    else
        Context.write(value)
    end if
end for
```

the computing model more flexible. The alignment task is performed in parallel on each node in the cluster. Start single-end alignment algorithm for single-end sequence and pair-end alignment algorithm for pair-end sequence on the node. The output is the result of sequence alignment in the partition on the node. After the alignment is completed, the results are uploaded to HDFS. When the alignment task of all nodes has been completed, multiple result files in HDFS are merged into a unified file, and the process ends.

6 Results and discussion

6.1 Experimental design

The following experiments were designed to verify the performance of our Gene Sequence Parallel Alignment Model. The gene data were extracted from the 1000 Genome Projects [18]. The 3.3G $GRCh38.p12$ was taken as the reference genome, while two datasets, $ERR000589$ and $SRR062634$, were selected as the short reads sequence. The specific information of the sequence is shown in Table 8.

Table 1: Short reads sequence datasets

Tag	Name	Number of reads	Read length (bp)	Size(GB)
D1	NA12750/ERR000589	1.2×10^7	51	5.2
D2	HG00096/SRR062634	6.7×10^6	200	3.5

As shown in Table 8, dataset D1 is composed of pair-end sequences with a length of 51bp. It is suitable for BWA backtrack algorithm. Dataset D2 is composed of single-end sequences with a length of 200 bp. It is suitable for BWA MEM algorithm. The two datasets differ in size, sequence length, sequencing method and alignment algorithm. In order to verify the universality and stability of the computing model for different data sets, the choice of experimental data should be representative [6,7]. Therefore, two data sets with different characteristics are selected in this experiment. The test cluster is a Hadoop cluster of one name node and eight data nodes. Each node is a VMware virtual machine with 8-core CPU, 8G memory and 1T hard disk. The Hadoop uses the version of 2.7.3. The operating system is Red Hat Enterprise Linux 6.5.

- Experiment 1, the BWA mapping was performed with D1 and D2 as inputs. The same computing tasks were run on Hadoop cluster with 1, 2, 4, 6, and 8 work nodes, respectively.

The time consumption, speedup and efficiency of each task were measured to evaluate the model performance.

- Experiment 2, the BWA mapping was performed with D1 and D2 as inputs. The same computing tasks were run on single node with 1, 2, 4, 6, and 8 threads, respectively. These time-consuming are compared with those of the same tasks on distributed clusters with 1, 2, 4, 6 and 8 nodes, respectively.

6.2 Results analysis

All tasks were completed smoothly. The experimental results were the same as those of single-machine operation, but achieved at a much shorter time. Table 2 shows the time consumption, speedup and efficiency of Experiment 1. As shown in Table 2, as the number of nodes increases, the time consumption of both tasks decreases dramatically. It shows that the model has good scalability, and the experiment time can be reduced by adding more nodes. The trend of speedup ratio shows that more nodes can make the acceleration effect more obvious. But it can't achieve linear acceleration, as shown in Figure 3. The reason is that when the number of nodes in the cluster increases, the overhead for cluster management will increase, and the overhead for computing task scheduling and resource management will also increase, resulting in a decrease in resource utilization. That's why the efficiency can't always be 1 [3,4]. The dataset D1 has shorter read length and lower computational complexity. Although the total data size exceeds D2, the speedup ratio is still slightly higher than D2. It is shown that the computing model has better speedup ratio for data sets with shorter read length. Experiment 1 proves that the parallel computing model of gene sequence alignment based on multiple inputs and outputs can greatly reduce the computing time for different data sets and different alignment algorithms, and has better speedup ratio and parallel computing efficiency.

Table 2: Performance comparison of Hadoop computing model

Content	Dataset	Number of nodes				
		1	2	4	6	8
Time consumption (m)	D1	258.8	131.1	67.8	45.3	38.5
	D2	249.6	127.1	72.8	49.5	36.7
Speedup	D1	1.0	2.0	3.8	5.7	6.7
	D2	1.0	2.0	3.4	5.0	6.8
Efficiency	D1	1.00	1.00	0.95	0.95	0.84
	D2	1.00	1.00	0.85	0.83	0.85

Table 3 shows the time-consuming of Experiment 2. Generally speaking, the execution time of both methods decreases with the increase of the number of nodes or threads. Figure 4 shows the trend of time-consuming. As can be seen from the Figure 4, the BWA multithread mode slows down the time-consuming after the start of four threads. However, Hadoop computing model still maintains a good reduction in processing time after 4 nodes. Experiment 2 indicates that the performance of BWA multithread mode is affected by single-node memory and CPU capacity. It cannot increase the speed of operation by increasing the number of threads blindly, and it is not scalable. The proposed computing model has good scalability. As long as there are enough work nodes, the computing time can be reduced to a lower level. Of course, the number of nodes cannot be increased indefinitely, because increasing the number of nodes will lead to a decrease in parallel efficiency, and the balance between the number of nodes and efficiency should be achieved [25].

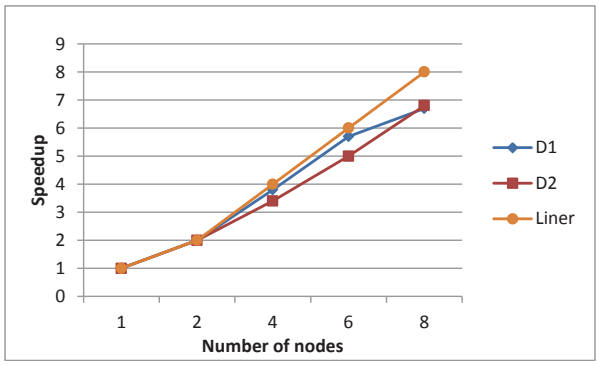


Figure 3: Speedup of Hadoop computing model with dataset D1 and D2

Table 3: Execution time of Hadoop and BWA multiple threads

Method	Dataset	Number of nodes/threads				
		1	2	4	6	8
Hadoop (m)	D1	258.8	131.1	67.8	45.3	38.5
	D2	249.6	127.1	72.8	49.5	36.7
Threads (m)	D1	258.8	154.6	127	124.6	123.8
	D2	249.6	102.5	52.9	53.2	51.8

The proposed computing model was further contrasted against several excellent parallel computing methods, which have been proved as capable of improving the performance of gene sequence alignment. All the experiments were performed using the same dataset on Hadoop clusters with different configurations. The grouping experiments were performed at 1, 2, 4, 6 and 8 nodes. Since these algorithms use different computing environments, the time consumption was not compared in the same dataset. In terms of the speedup ratio of parallel computing (Figure 5), the proposed computing model had certain advantages over the contrastive algorithms. The results show that gene sequence multiple input and output method both saves the time of data preprocessing and reduces the burden of MapReduce computing model, improving the efficiency of parallel computing.

The multiple inputs and outputs method on Hadoop platform can effectively process single-end and pair-end sequences of gene data, which makes the operation of gene sequences not limited to single file and avoids the use of third-party tools in parallel computing of gene sequence

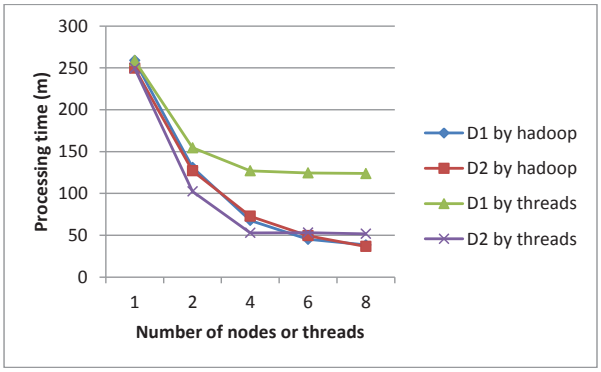


Figure 4: Execution time comparison between Hadoop and BWA multiple threads

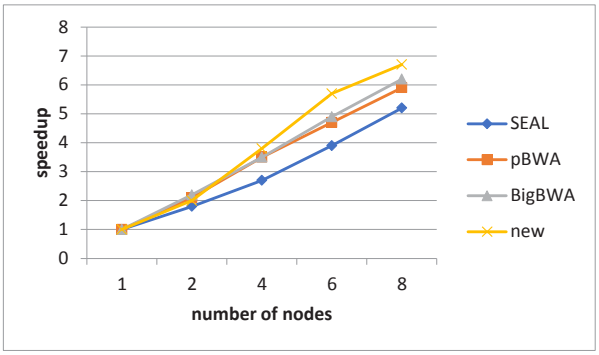


Figure 5: Speedup comparison of different parallel BWA

alignment. The parallel computing model of gene sequence alignment based on multi input and output uses Hadoop API to complete the task of gene sequence alignment, including the data input, data distribution, distributed computing and result processing, which ensures the uniformity of application. Experiments show that this computing model makes the task of gene sequence alignment scalable in distributed cluster. Compared with single node algorithm, the computing time of the same task is significantly reduced. Compared with multi-threaded parallel computing mode and other parallel gene data computing schemes, this computing model has certain advantages.

Funding

This work was supported by National Natural Science Foundation of China project 61462070.

Author contributions. Conflict of interest

The authors contributed equally to this work. The authors declare no conflict of interest.

Bibliography

[1] Abuin, J.M.; Pichel, J.C.; Pena, T.F.; Amiqo, J. (2015). BigBWA: Approaching the Burrows-Wheeler Aligner to Big Data Technologies, *Bioinformatics*, 31(24), 4003-4005, 2015.

[2] Almeida, J.S.; Gruneberg, A.; Maass, W.; Vinga, S. (2012). Fractal MapReduce decomposition of sequence alignment, *Algorithms for Molecular Biology*, 7(1), 1-12, 2012.

[3] Bala, R.J.; Govinda, R.M.; Murthy, C.S.N. (2018). Reliability analysis and failure rate evaluation of load haul dump machines using Weibull distribution analysis, *Mathematical Modelling of Engineering Problems*, 5(2), 116-122, 2018.

[4] Chen, Z.; Hou, Z.W.; Yang, Q.Q.; Chen, X.B. (2018). Adaptive Meshing Based on the Multi-level Partition of Unity and Dynamic Particle Systems for Medical Image Datasets, *International Journal Bioautomation*, 22(3), 229-238, 2018.

[5] Cock, P.J.; Fields, C.J.; Goto, N.; Heuer, M.; Rice, P.M. (2009). The Sanger FASTQ file format for sequences with quality scores and the Solexa/Illumina FASTQ variants, *Nucleic Acids Research*, 38(6), 1767-1771, 2009.

[6] Dai, Y.; Wu, W.; Zhou, H.B.; Zhang, J.; Ma, F.Y. (2018). Numerical Simulation and Optimization of Oil Jet Lubrication for Rotorcraft Meshing Gears, *International Journal of Simulation Modelling*, 17(2), 318-326, 2018.

[7] Dai, Y.; Zhu, X.; Zhou, H. ; Mao, Z. ; Wu, W. (2018). Trajectory Tracking Control for Seafloor Tracked Vehicle By Adaptive Neural-Fuzzy Inference System Algorithm, *International Journal of Computers Communications & Control*, 13(4), 465-476, 2018.

[8] Dean, J.; Ghemawat, S. (2004). MapReduce: Simplified Data Processing on Large Clusters. Proceedings of Sixth Symposium on Operating System Design and Implementation (OSD2004), *USENIX Association*, 2004.

[9] Decap, D.; Reumers, J.; Herzeel, C.; Costanza, P.; Fostier, J. (2015). Halvade: scalable sequence analysis with MapReduce, *Bioinformatics*, 31(15), 2482-2488, 2015.

[10] Gufler, B.; Augsten, N.; Reiser, A.; Kemper, A. (2012). The Partition Cost Model for Load Balancing in MapReduce, *Cloud Computing and Services Science*, Springer New York, 371-387, 2012.

[11] Li, H. (2013). Aligning sequence reads, clone sequences and assembly contigs with BWA-MEM, *Genomics*, 1-3, 2013.

[12] Li, H. (2009). The Sequence Alignment / Map (SAM) Format, *Bioinformatics*, 25(1-2), 1653-1654, 2009.

[13] Metzker, M.L. (2010). Sequencing technologies - the next generation, *Nature Reviews Genetics*, 11(1), 31-46, 2010.

[14] Pandey, R.V.; Schlotterer, C. (2013). DistMap: A Toolkit for Distributed Short Read Mapping on a Hadoop Cluster, *PLOS ONE*, 8(8), e72614, 2013.

[15] Pireddu, L.; Leo, S.; Zanetti, G. (2011). SEAL: a distributed short read mapping and duplicate removal tool, *Bioinformatics*, 27(15), 2159-2160, 2011.

[16] Schatz, M.C. (2009). CloudBurst: highly sensitive read mapping with MapReduce, *Bioinformatics*, 25(11), 1363-1369, 2009.

[17] Taylor, R.C. (2010); An overview of the Hadoop/MapReduce/HBase framework and its current applications in bioinformatics, *Bmc Bioinformatics*, 11(S12), S1, 2010.

[18] Watson, J.D. (1990). The Human Genome Project: Past, Present, and Future, *Science*, 248(4951), 44-49, 1990.

[19] Zhang, J.; Wu, Y.Q.; Yi, H.C. (2018). Forward modelling of circular loop source and calculation of whole area apparent resistivity based on TEM, *Traitement du Signal*, 35(2), 183-198, 2018.

[20] [Online]. Available: hadoop.apache.org/, Accesed on 20 June 2018.

INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL
ISSN 1841-9836, e-ISSN 1841-9844, 14(2), 154-169, April 2019.

Weighted Random Search for Hyperparameter Optimization

A.C. Florea, R. Andonie

Adrian-Cătălin Florea*
Department of Electronics and Computers
Transilvania University of Braşov, Romania
*Corresponding author: acflorea@unitbv.ro

Răzvan Andonie
Department of Computer Science
Central Washington University, USA
andonie@cwu.edu

Abstract: We introduce an improved version of Random Search (RS), used here for hyperparameter optimization of machine learning algorithms. Unlike the standard RS, which generates for each trial new values for all hyperparameters, we generate new values for each hyperparameter with a probability of change. The intuition behind our approach is that a value that already triggered a good result is a good candidate for the next step, and should be tested in new combinations of hyperparameter values. Within the same computational budget, our method yields better results than the standard RS. Our theoretical results prove this statement. We test our method on a variation of one of the most commonly used objective function for this class of problems (the Grievank function) and for the hyperparameter optimization of a deep learning CNN architecture. Our results can be generalized to any optimization problem defined on a discrete domain.
Keywords: Hyperparameter optimization, random search, deep learning, convolutional neural network.

1 Introduction

The vast majority of machine learning algorithms involve two different sets of parameters: the training parameters and the meta-parameters (also known as *hyperparameters*). While the training parameters are learned during the training phase, the values of the hyperparameters have to be specified before the learning phase. For instance, the hyperparameters of neural networks typically specify the architecture of the network (number and type of layers, number and type of nodes, etc).

Determining the optimal combination of hyperparameter values leading to the best generalization performance can be done through repeated training and evaluation sessions, trying different combinations of hyperparameter values. We call each training + evaluation process for one combination of hyperparameter values a *trial*. Each trial is computationally expensive, since it involves re-training the model. In addition, the number of trials increases generally exponential with the number of hyperparameters. Therefore, it is important to reduce the number or trials [9]. This can be done by both reducing the number of hyperparameters and reducing the value range of each hyperparameter, while still maximizing the probability to hit the optimal combination [2, 3].

Various hyperparameter optimization methods were developed during the years, ranging from very simple ones, such as Grid Search (GS) and manual tuning [14, 20, 28][1], to highly

[1]https://github.com/jaak-s/nips2014-survey - 82 out of 86 optimization related papers presented at the NIPS 2014 conference used GS.

elaborated techniques: Nelder-Mead [1, 24], simulated annealing [17], evolutionary algorithms [12], Bayesian methods [32], etc.

Recently, there has been significant interest in the area of hyperparameter optimization, especially since the rise of deep learning which puts a lot of pressure on the existing techniques due to the very large number of hyperparameters involved and the significant training time needed for such architectures. The focus in hyperparameter optimizations presently oscillates between introducing more sophisticated techniques (Sequential Model-Based Global Optimization [2], reinforcement learning [34,35], etc) and various attempts to optimize existing simple techniques.

RS falls into the category of simple algorithms [2, 3]. Making use of the same computational budget, RS generally yields better results than GS or more complicated hyperparameter optimization methods [2]. Especially in higher dimensional spaces, the computation resources required by RS methods are significantly lower than for GS [21]. RS consists in drawing samples from the parameter space following a particular distribution for each of the parameters. Each trial is drawn and evaluated independently from the others, which makes RS a very good candidate for parallel implementations.

Some recent attempts to optimize the RS algorithm are: Li's *et al.* Hyperband [22], which speeds up RS through adaptive resource allocation and early-stopping; Domhan *et al.* [8], which have developed a probabilistic model to mimic early termination of sub-optimal candidate; and Florea *et al.* [9], where we introduced a dynamically computed stopping criterion for RS, reducing the number of trials without reducing the generalization performance.

There are various software libraries implementing hyperparameter optimization methods. Hyperopt [4] and Optunity [7] are currently two of the most advanced standalone packages. Bayesian techniques are implemented by packages like BayesianOptimization [29] and pyGPGO [27]. Some of the best known general purpose machine learning software libraries also provide hyperparameter optimization: LIBSVM [5] and scikit-learn [26] come with their own implementation of GS, with scikit-learn also offering support for RS. Auto-WEKA [18], built on top of Weka [11] is able to perform GS, RS, and Bayesian optimization.

Lately, commercial cloud-based services started to offer hyperparameter optimization capabilities. Among them we count Google HyperTune [38], BigML's OptiML [36], and SigOpt [40]. All of them support mixed search domains, SigOpt being able to handle multi-objective, multi-solution, constraint (linear and black-box), and parallel optimization.

In this context, our contribution is an improved version of the RS method, the *Weighted Random Search* (WRS) method. Unlike the standard RS, which generates for each trial new values for all hyperparameters, we generate new values for each hyperparameter with a probability of change p and we use the best value found so far for that particular hyperparameter with probability $1 - p$, where p is proportional to the hyperparameter's relative importance in the variation of the objective function. The intuition behind our approach is that a value that already triggered a good result is a good candidate for a new trial and should be tested in new combinations of hyperparameter values.

For the same number of trials, the WRS algorithm produces significantly better results than RS. We obtained theoretical results which prove this statement. We tested our algorithm on a slightly modified version of one of the most commonly used objective function for this class of problems - the Grievank [10] function, as well as for the hyperparameter optimization of a deep learning CNN architecture using the CIFAR-10 [37] dataset.

Unlike our previous work on RS optimization [9], where our focus was on the dynamic reduction of the number of trials, the focus of the WRS method is the optimization of the classification (prediction) performance within the same computational budget. The two approaches make use of different optimization techniques.

The paper proceeds as follows. Section 2 is a general presentation of our WRS algorithm.

Section 3 describes theoretical results and the convergence of WRS. Sections 4 and 5 contain experimental results. The paper is concluded with Section 6.

2 The WRS method

We first present the generic intuitive description of the WRS algorithm, which is the core of our contribution. Technical details will be provided later.

The standard RS technique [2] generates a new multi-dimensional sample at each step k, with new random values for each of the sample's dimensions - features, in our case - $X^k = \{x_i^k\}, i = 1, \ldots, d$, where x_i is generated according to a probability distribution $P_i(x), i = 1, \ldots, d$, and d is the number of dimensions.

WRS is an improved version of RS, designed for hyperparameter optimization. It assigns probabilities of change $p_i, i = 1, \ldots, d$ to each dimension. For each dimension i, after a certain number of steps k_i, instead of always generating a new value, we generate it with probability p_i and use the best value known so far with probability $1 - p_i$.

The intuition behind the proposed algorithm is that after already fixing d_0 $(1 < d_0 < d)$ values, each d-dimensional optimization problem reduces itself to a $d - d_0$ dimensional one. In the context of this $d - d_0$ dimensional problem, choosing a set of values that already performed well for the remaining dimensions might prove more fruitful than choosing some $d - d_0$ random values. In order to avoid getting stuck in a local optimum, instead of setting a hard boundary between choosing the best combination of values found so far or generating new random samples, we assign probabilities of change for each dimension of the search space.

WRS has two phases. In the first phase, it runs the RS for a predefined number of trials and allows: *a)* to identify the best combination of values so far; and *b)* to give enough input on the importance of each dimension in the optimization process. The second phase considers the probabilities of change and generates the candidate values according to them. Between these two phases, we run one instance of fANOVA [15], in order to determine the importance of each dimension with respect to the objective function. Intuitively, the most important dimension (the dimension that yields the largest variation of the objective function) is the one that should change most frequently, to cover as much of the variation range as possible. For a dimension with small variation of the objective function, it might be more efficient to keep a certain temporary optimum value once this has been identified.

A step of the WRS algorithm applied to function maximization is described by Algorithm 1, whereas the entire method is detailed in Algorithm 2. F is the objective function, the value $F(X)$ has to be computed for each argument, X^k is the best argument at iteration k, whereas N is the total number of iterations.

At each step of Algorithm 2, at least one dimension will change, hence we always choose at least one of the p_i probabilities to be equal to one. For the other probabilities, any value in $(0, 1]$ is valid. If all values are one, then we are in the case of RS.

Besides a way to compute the objective function, Algorithm 1 requires only the combination of values that yields the best $F(X)$ value obtained so far and the probability of change for each dimension. The current optimal value of the objective function can be made optional, since the comparison can be done outside of Algorithm 1. Algorithm 2 coordinates the sequence of the described steps and calls Algorithm 1 in a loop, until the maximum number of trials N is reached.

Algorithm 1 A WRS Step - Objective Function Maximization

Input: F; $(X^k, F(X^k))$; $p_i, k_i, P_i(x), i = 1, \ldots, d$
Output: $(X^{k+1}, F(X^{k+1}))$

Randomly generate p, uniform in (0,1)
for $i = 1$ to d **do**
  if ($p_i \geq p$ or $k \leq k_i$) **then**
    // either the probability condition is met or more samples are needed
    Generate x_i^{k+1} according to $P_i(x)$
  else
    $x_i^{k+1} = x_i^k$
  end if
end for
// usually this is the most time consuming step
Compute $F(X^{k+1})$
if $F(X^{k+1}) \geq F(X^k)$ **then**
  return $(X^{k+1}, F(X^{k+1}))$
else
  return $(X^k, F(X^k))$
end if

Algorithm 2 WRS - Objective Function Maximization

Input: F; N; $P_i(x), i = 1, \ldots, d$
Output: $(X^N, F(X^N))$

// Phase 1 - Run RS
for $k = 1$ to $N_0 < N$ **do**
  Perform RS step, compute $(X^k, F(X^k))$
end for
// Intermediate phase, determine input for WRS
Determine the probability of change $p_i, i = 1, \ldots, d$
Determine the minimum number of required values $k_i, i = 1, \ldots, d$
// Phase 2 - Run WRS
for $k = N_0 + 1$ to N **do**
  Perform WRS Step described in Algorithm 1, compute $(X^k, F(X^k))$
end for
return $(X^N, F(X^N))$

3 Theoretical aspects and convergence

We aim to analyze the theoretical convergence of Algorithm 2 and compare it to the RS method. Similar to GS and RS, we make the assumption that there is no statistical correlation between the variables of the objective function (hyperparameters). To make explanations more intuitive, we first discuss the two-dimensional case, and then generalize for the multi-dimensional case. We will also define what we consider "a set of good candidate values" for p_i and $k_i, i = 1, \ldots, d$ (used in steps 6 & 7, Algorithm 2). We denote by $n \geq 1$ the number of iterations (steps) for both RS and WRS.

3.1 Two-dimensional case

In the two-dimensional case ($d = 2$), we aim to maximize a function $F : \mathrm{S}_1 \times \mathrm{S}_2 \to \mathbb{R}$, where S_1 and S_2 are countable sets. We define as *global optimum* the point $X^*(x_1^*, x_2^*)$, with $x_1^* \in \mathrm{S}_1$ and $x_2^* \in \mathrm{S}_2$, so that $F(X^*) \geq F(X), \forall X \in \mathrm{S}_1 \times \mathrm{S}_2$. $p_i, k_i, (i = 1, 2)$ are the probabilities of change, respectively, the required number of distinct values for x_i, as previously defined. $|S_i|$ is the cardinality of $S_i, i = 1, 2$. We denote by $p_{RS:n}$ and $p_{WRS:n}$ the probability for RS, respectively WRS, to reach the global optimum after n steps.

The following theorem establishes that, in the two-dimensional case we can choose k_2 so that

$$p_{WRS:n} \geq p_{RS:n} \tag{1}$$

Theorem 1. *For any function $F : S_1 \times S_2 \to \mathbb{R}$ there exists k_2, so that $p_{WRS:n} \geq p_{RS:n}$.*

Proof: We consider the case of maximizing function F, and choose the arguments in the decreasing order of their probabilities of change. Since the value for one dimension always changes, we have $p_1 = 1, p_2 \leq 1$. Having $p_1 = 1$, the value of k_1 can be ignored: the condition at step 3 in Algorithm 1 will be always true for $i = 1$.

At each step k, $k \leq k_2$, WRS is identical with RS and we have $p_{WRS:k} = p_{RS:k}$. At step $k+1 > k_2$, RS generates new values for x_1^{k+1} and x_2^{k+1}, and computes $F(x_1^{k+1}, x_2^{k+1})$. For WRS, x_1^{k+1} is generated with probability one, but x_2^{k+1} is generated with $p_2 \leq 1$. With probability $1 - p_2$, the best value known so far for x_2 is used, instead of generating a new one. X^{k+1} can be written as:

$$X^{k+1} = \begin{cases} (x_1^{k+1}, x_2^{k+1}), & \text{with probability } p_2 \\ (x_1^{k+1}, x_2^{k}), & \text{with probability } 1 - p_2 \end{cases} \tag{2}$$

With probability p_2, each step in WRS is identical to the same step in RS, and all points in $\mathrm{S}_1 \times \mathrm{S}_2$ are accessible to WRS. Therefore, RS and WRS have the same search space and both converge probabilistically to the global optimum.

Ignoring the statistical correlation between the two variables, the probability of RS to hit the optimum after one iteration (the best case) is:

$$p_{RS} = \frac{1}{|S_1|} \frac{1}{|S_2|} \tag{3}$$

For WRS, this probability is:

$$p_{WRS} = \frac{1}{|S_1|} \left(p_2 \frac{1}{|S_2|} + (1 - p_2) \frac{1}{|S_2| - m_2 + 1} \right) \tag{4}$$

where m_2 is the number of distinct values already generated for x_2.

Using (3) and (4), (1) becomes:

$$1-(1-p_{WRS})^n \geq 1-(1-p_{RS})^n \tag{5}$$

which is equivalent to

$$\frac{1}{|S_1|}\Big(p_2\frac{1}{|S_2|}+(1-p_2)\frac{1}{|S_2|-m_2+1}\Big) \geq \frac{1}{|S_1|}\frac{1}{|S_2|} \tag{6}$$

After multiplying both sides by $|S_1|$, (6) can be rewritten as

$$p_2\frac{1}{|S_2|}+(1-p_2)\frac{1}{|S_2|-m_2+1} \geq \frac{1}{|S_2|} \tag{7}$$

which reduces to

$$p_2(1-m_2) \geq 1-m_2 \tag{8}$$

Because $p_2 \leq 1$, (8) is true for $m_2 > 1$. Relation (1) is therefore satisfied if we choose k_2 so that at least two distinct values are generated for x_2.

□

3.2 Multi-dimensional case

For the general case of optimizing a function $F : \mathrm{S}_1 \times \mathrm{S}_2 \ldots \times \mathrm{S}_d \to \mathbb{R}$, with $S_i, i = 1, \ldots, d$ countable sets and under the same assumption that the variables are not statistically correlated, P_{RS} and P_{WRS} are defined as:

$$p_{RS} = \prod_{i=1}^{d}\frac{1}{|S_i|} \tag{9}$$

$$p_{WRS} = \frac{1}{|S_1|}\prod_{i=2}^{d}\Big(p_i\frac{1}{|S_i|}+(1-p_i)\frac{1}{|S_i|-m_i+1}\Big) \tag{10}$$

where m_i is the number of distinct values already generated for x_i.

Following the rationale from Section 3.1, we have the following theorem:

Theorem 2. *For any function $F : S_1 \times S_2 \ldots \times S_d \to \mathbb{R}$ there exist $k_i, i = 1, \ldots, d$, so that $p_{WRS:n} \geq p_{RS:n}$.*

Proof:

We consider again the maximization of function F.

Given $k_i, i = 1, \ldots, d$ the minimum number of values required for each of the dimensions x_i with $k_i \leq k_{i+1}, i = 1, \ldots, d-1$ and $k \geq k_d$, X_{k+1} is given by:

$$\begin{cases} (x_1^{k+1}, x_2^{k+1}, \ldots, x_{d-1}^{k+1}, x_d^{k+1}), & \text{with probability } p_d \\ (x_1^{k+1}, x_2^{k+1}, \ldots, x_{d-1}^{k+1}, x_d^{k}), & \text{with probability } p_{d-1}-p_d \\ \ldots & \\ (x_1^{k+1}, x_2^{k+1}, \ldots, x_{d-1}^{k}, x_d^{k}), & \text{with probability } p_2-p_3 \\ (x_1^{k+1}, x_2^{k}, \ldots, x_{d-1}^{k}, x_d^{k}), & \text{with probability } 1-p_2 \end{cases} \tag{11}$$

Starting from (9) and (10), we can express $P_{WRS:n}$ as:

$$1-\left(1-\frac{1}{|S_1|}\prod_{i=2}^{d}\Big(p_i\frac{1}{|S_i|}+(1-p_i)\frac{1}{|S_i|-m_i+1}\Big)\right)^n \tag{12}$$

and $P_{RS:n}$ as:

$$1 - \Big(1 - \prod_{i=1}^{d} \frac{1}{|S_i|}\Big)^n \tag{13}$$

Since all elements of the products from (12) and (13) are positive ($1 - p_i \geq 0$, and m_i cannot be greater than $|S_i|$), a sufficient condition to satisfy (1) is:

$$\Big(\frac{1}{|S_i|} + (1 - p_i)\frac{1}{|S_i| - m_i + 1}\Big) \geq \frac{1}{|S_i|} \tag{14}$$

for each $i \geq 2$), which reduces to

$$p_i(1 - m_i) \geq 1 - m_i \tag{15}$$

and, since $p_i \leq 1$, is equivalent with

$$m_i \geq 2, \text{for } i = 2, \ldots, d \tag{16}$$

Relation (1) is satisfied if we choose k_i so that at least two distinct values are generated for each dimension.

□

According to these results, for a well chosen set of $k_i, i = 1, \ldots, d$, at any step n, WRS has a greater probability than RS to find the global optimum. Therefore, given the same number of iterations, on average, WRS finds the global optimum faster than RS. In other words, on average, WRS converges faster than RS.

Moreover, for WRS, the number of generated values for $x_i, i = 1, \ldots, d$, follows a binomial distribution with probability p_i. After n steps, the expected value for this distribution is np_i. Therefore, m_i has, on average, an upper bound of np_i. The number of distinct generated values depends on the cardinality of S_i and the probability distribution used to generate x_i.

For example, in the case of the uniform distribution, the expected value for m_i is:

$$E[m_i] = \sum_{1}^{|S_i|} \left(1 - \left(\frac{|Si| - 1}{|Si|}\right)^{np_i}\right) \tag{17}$$

and $m_i > 1$ when $np_i > 1$. Hence, for any number of steps n, with $n \geq 1/p_i$, (1) is true. By choosing k_i so that $k_i > 1/p_i$, (1) is true for all values of n. It can be also observed that the difference between $p_{WRS:n}$ and $p_{RS:n}$ increases with an increasing value of n.

3.3 Choosing p_i and k_i

Regardless of the distribution used for generating x_i, by choosing for k_i (step 6, Algorithm 2) a value that can guarantee the generation of at least two distinct samples, (1) is true and WRS has a higher probability to find the optimum than RS.

We decide to sort the function variables depending on their importance (weight) and assign their probabilities p_i accordingly: the smaller the weight of a parameter, the smaller it's probability of change. Therefore, the most important parameter is the one that will always change ($p_1 = 1$). In order to compute the weight of each parameter, we run RS for a predefined number of steps, $N_0 < N$. On the obtained values, we apply fANOVA [15] to estimate the importance of the hyperparameters. If w_i is the weight of the i-th parameter and w_1 is the weight of the most important one, then $p_i = w_i/w_1, i = 1, \ldots, d$.

By assigning higher probabilities of change to the most important parameters and running RS for N_0 steps, we make sure that (16) is satisfied for these parameters. For simplicity, we set $k_i = N_0$ for all parameters, but these values can be adjusted depending on the objective function.

4 An example: Griewank function optimization

To illustrate the concept behind WRS, we consider a simple function with a known analytic form. Since the function is very fast to compute, we can test the performance of our algorithm on a very large number of runs. This will allows us to perform an unpaired t-test on the results and rule out the random factor when assessing its performance.

The Grievank [10] function is widely used to test the convergence of optimization algorithms. It's analytic form is given by:

$$G_d = 1 + \frac{1}{4000}\sum_{i=1}^{d} x_i^2 - \prod_{i=1}^{d} \cos\frac{x_i}{\sqrt{i}} \tag{18}$$

The function poses a lot of stress on optimization algorithms due to its very large number of local minimums. We use a slightly modified version of G_6, given by:

$$G_6^* = 1 + \frac{i-1}{4000}\sum_{i=1}^{6} x_i^2 - \prod_{i=1}^{6} \cos\frac{x_i}{\sqrt{i}} \tag{19}$$

and maximize $-G_6^*$. The function has a global maximum at 0, for $x_i = 0, i = 1, \ldots, 6$. The term $i-1$ is introduced in order to alter the parameters' importance(weight) which, otherwise, would have been the same across all dimensions. We use $S = [-600, 600]$ for all six parameters and run the optimizer for 1000 trials, with an initial RS phase of $1000/e = 368$ steps [9]. After the first RS phase, we run fANOVA and obtain the weights of the parameters, listed along with their probabilities of change in Table 1.

Table 1: Parameter weights and probabilities for G_6^*

Parameter	x_1	x_2	x_3	x_4	x_5	x_6
Weight	0.07	0.18	1.24	7.77	23.52	43.96
Probability	0.002	0.004	0.028	0.177	0.535	1.00

We compare our results against RS, on the same search space, performing 1000 trials on 10000 runs. Table 2 shows the best result achieved by both RS and WRS across all 10000 runs, as well as the average value and the standard deviation of the achieved results across all runs. The standard error for the t-test is 0.176, $df = 19998$ and P-value ≤ 0.001.

Table 2: WRS vs. RS results for G_6^* - values for 1000 runs

Optimizer	Best Found Value	Average Value	SD
RS	-1.50	-33.10	14.06
WRS	**-1.28**	**-14.58**	**10.63**

The results obtained by WRS are clearly better than the ones achieved by RS, as also depicted in Fig. 1.

Fig. 2 shows the results obtained for one optimization session with 1000 trials. It can be observed that the algorithm tends to achieve improving results as the number of trials increases.

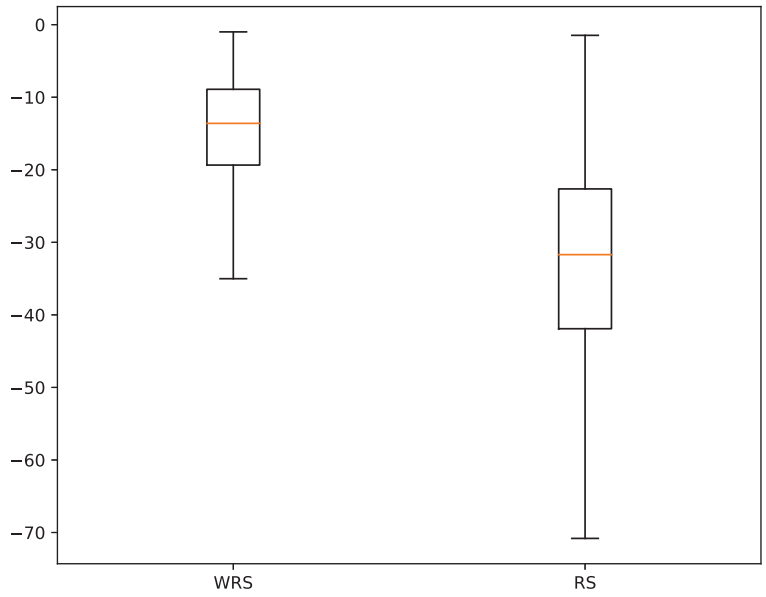


Figure 1: Performance of WRS vs. RS for the G_6^* optimization

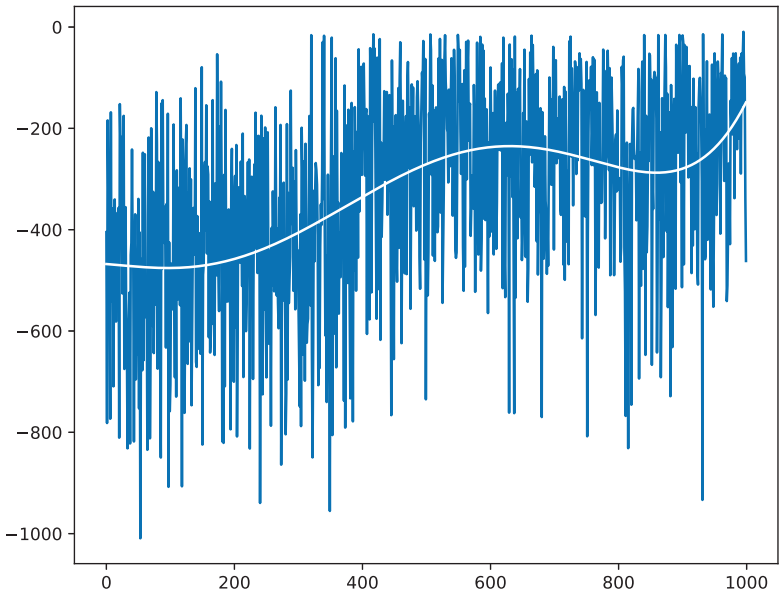


Figure 2: Convergence of WRS for the G_6^* function

5 CNN hyperparameter optimization

Our next application of the WRS is for the optimization of a CNN architecture. Currently CNN in one of the best and most used tools for image recognition and machine vision [25] and there has been a lot if interest in developing optimal CNN architectures [13, 19, 31, 33]. Current CNN architectures are complex, with a high number of hyperparameters. In addition, the training sets for CNNs are large and this increases training times. Hence, we have a high number of trials, each trail with significant execution time. Decreasing the number of trials is critical.

When applying WRS to our CNN optimization problem we consider the following hyperparameters:

- The number of convolution layers - an integer value in the set $\{3, 4, 5, 6\}$;
- The number of fully-connected layers - an integer value in the set $\{1, 2, 3, 4\}$;
- The number of output filters in each convolution layer - an integer value in the range $[100, 1024]$;
- The number of neurons in each fully connected layer - an integer value in the range $[1024, 2048]$.

We generate each hyperparameter according to the uniform distribution and assess the performance of the model solely by the classification accuracy.

We use Keras [6] to train and test the CNN for 300 trials - ten epochs each - on the CIFAR-10 [37] dataset. We run our test on an IBM S822LC cluster with IBM POWER8 nodes, NVLink and NVidia Tesla P100 GPUs[2]. The CIFAR-10 dataset consists of 60000 32×32 color images in 10 classes, with 6000 images per class. The data is split into 50000 training images and 10000 test images. We do not use data augmentation.

The base architecture of the network is represented in Fig. 3. The model has between three and six 3×3 convolutional layers and between one and four fully connected layers. Both the convolutional and fully connected layers use ReLU [23] activation and the output layer uses softmax. We add one 2×2 MAX pooling layer with a dropout [25] of 0.25 for every two convolutional layers and use a dropout of 0.5 for the fully connected layers. We compare the results obtained by our WRS algorithm against the ones obtained by the RS, Nelder-Mead (NM), Particle Swarm (PS) [16] and Sobol Sequences (SS) [30] implementations provided by Optunity [39].

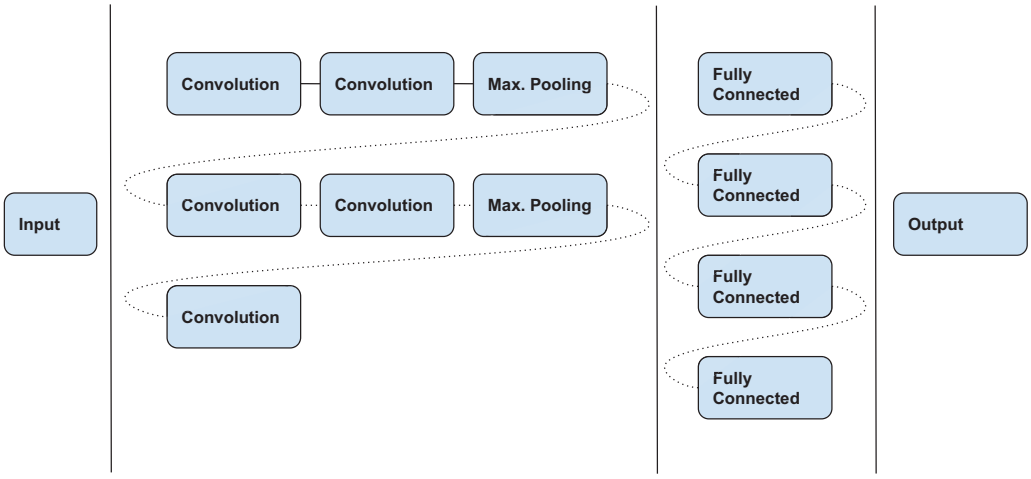


Figure 3: The CNN architecture

After the first phase of the algorithm, which consists in running RS for $300/e = 110$ trials, we obtain the weights for each parameter. These values, along with the probabilities of change, are listed in Table 3. After running fANOVA, the resulted most important three parameters are (in decreasing order of their weights): the number of neurons in the first fully connected layer,

[2]http://www.cwu.edu/faculty/turing-cwu-supercomputer

the number of fully connected layers, and the number of convolutional layers. The weights of the other parameters are more than an order of magnitude smaller. Therefore, the second phase of WRS clearly favors the change in the first three most important parameters.

Table 3: Parameter weights and probabilities for CNN

Convolutional Layers	Fully Connected Layers	Conv 1	Conv 2	Conv 3	Conv 4	Conv 5	Conv 6	Full 1	Full 2	Full 3	Full 4
7.4	11.85	0.51	0.79	1.62	0.73	2.26	1.26	26.28	0.87	3.22	1.75
0.28	0.45	0.02	0.03	0.06	0.03	0.09	0.05	1.00	0.03	0.12	0.07

Fig. 4 shows the least squares five degree polynomial fit on the accuracy results obtained for each of the 300 trials using: WRS - the solid line, RS, NM, PS, SS - the dashed lines. The trend of the WRS performance is similar to the one from Fig. 1. The plot considers the actual values, reported at each iteration, instead of the local best in order to better reveal the variation of those values.

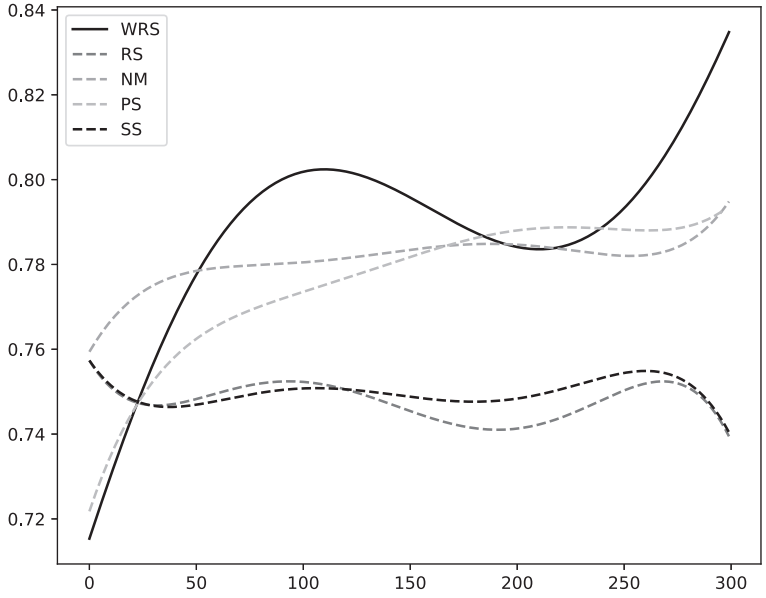


Figure 4: Least squares five degree polynomial fit on RS, NM, PS, SS vs. WRS accuracy for CIFAR-10 on 300 trials. The plot considers the values reported at each iteration

The best accuracy, as well as the average and standard deviation, across all 300 trials for all algorithms, are depicted in Table 4. WRS method outperforms all other considered methods (see Table 4 and Fig. 5).

Table 4: Algorithms' results for CNN accuracy on CIFAR-10

Optimizer	Best Result	Average	SD
WRS	**0.85**	**0.79**	0.09
RS	0.81	0.75	0.04
NM	0.81	0.77	0.03
PS	0.83	0.78	0.03
SS	0.82	0.75	0.05

Table 5 shows the best found architecture by each algorithm. We observe that for the

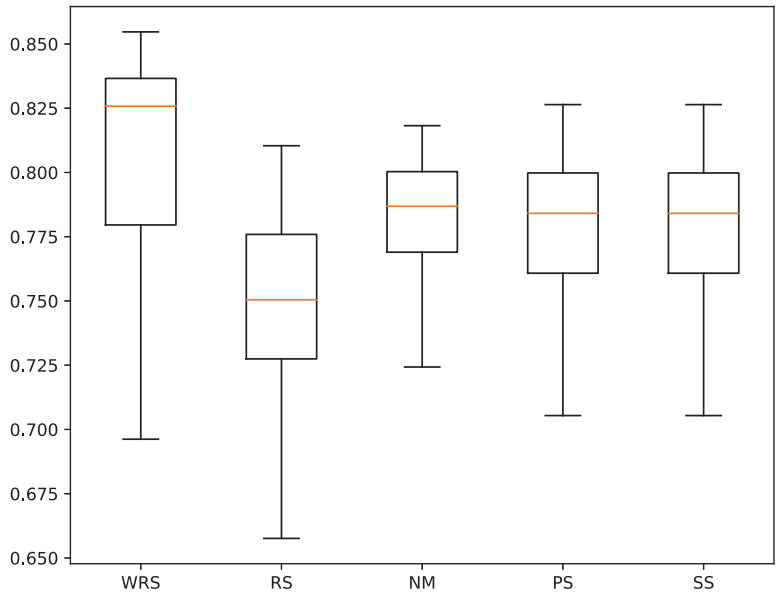


Figure 5: Performance of WRS, RS, NM, PS and SS for CNN optimization

WRS and RS methods, the resulted architectures have only one fully connected layer and several convolutional layers (five for RS, six for WRS).

Table 5: Best identified CNN architectures on CIFAR-10

Optimizer	Convolutional Layers	Fully Connected Layers	Conv 1	Conv 2	Conv 3	Conv 4	Conv 5	Conv 6	Full 1	Full 2	Full 3	Full 4
WRS	6	1	736	508	664	916	186	352	1229	-	-	-
RS	5	1	876	114	892	696	617	-	1828	-	-	-
NM	5	3	564	564	564	560	563	-	1529	1542	1542	-
PS	5	1	479	792	584	411	593	-	1379	-	-	-
SS	5	2	402	933	750	997	777	-	1545	1268	-	-

Table 6: WRS Accuracy Average and Standard Deviation. Row headings are numbers of fully connected layers while column headings are numbers of convolutional layers

FC /C	1	2	3	4
3	0.74 (0.02)	0.70 (0.03)	0.74 (0.01)	0.69 (0.03)
4	0.78 (0.01)	0.74 (0.03)	0.74 (0.03)	0.63 (0.07)
5	0.81 (0.02)	0.80 (0.02)	0.74 (0.07)	0.65 (0.06)
6	**0.82 (0.01)**	0.76 (0.04)	0.72 (0.09)	0.39 (0.21)

Table 6 details the results obtained by WRS, showing the accuracy average and the standard deviation values for each combination: (number of fully connected layers, number of convolutional layers). Table 7 shows the number of trials performed by WRS for each of these combinations.

We notice that the WRS algorithm favors one of the combinations, namely {1, 6}, and uses it for almost two thirds of the number of trials. It is important to mention that within the best 200 trials, only 10 sets of values contain a different combination than {1, 6}. This is either {1, 5} - seven times, or {2, 5} - three times. The first different combination than {1, 6} is at the 136-th position. In Table 6, we observe that this combination also triggers the best results.

This, together with the fact that WRS performs on average better than RS, validates our

Table 7: WRS Number of Trials. Row headings are numbers of fully connected layers while column headings are numbers of convolutional layers

FC /C	1	2	3	4
3	4	4	4	7
4	8	3	8	9
5	9	7	9	4
6	**199**	6	10	9

hypothesis that the probability that this combination of hyperparameters corresponds to the global optimum is higher than for any other combination.

6 Conclusions

We have introduced an improved version of RS, the WRS method. Within the same computational budget (i.e., for the same number of iterations), WRS converges on average faster than RS. The WRS algorithm yields better results both for the optimization of a well known difficult mathematical function and for a CNN hyperparameter optimization problem. There is little information required to be transferred between the consecutive steps of the algorithm, as pointed out in the description of Algorithm 1. This implies that the WRS algorithm can be easily implemented in parallel. Since we made no assumptions on the objective function, our results can be generalized to other optimization problems defined on a discrete domain. We plan to test out algorithm on other classes of optimization problems, in particular on the optimization of various machine learning algorithms. We also plan to compare the results obtained with WRS with other more complicated optimization techniques, especially from the very promising area of Bayesian optimization.

Author contributions. Conflict of interest

The authors contributed equally to this work. The authors declare no conflict of interest.

Bibliography

[1] Albelwi, S.; Mahmood, A. (2017). A framework for designing the architectures of deep convolutional neural networks. *Entropy*, 19, 6 (2017).

[2] Bergstra, J.; Bardenet, R.; Bengio, Y.; Kegl, B. (2011). Algorithms for hyper-parameter optimization. In *NIPS* (2011), J. Shawe-Taylor, R. S. Zemel, P. L. Bartlett, F. C. N. Pereira, and K. Q. Weinberger, Eds., 2546–2554, 2011.

[3] Bergstra, J.; and Bengio, Y. (2012). Random Search for Hyper-Parameter Optimization. *Journal of Machine Learning Research*, 13, 281–305, 2012.

[4] Bergstra, J.; Komer, B.; Eliasmith, C.; Yamins, D.; and Cox, D. D. (2015). Hyperopt: a Python library for model selection and hyperparameter optimization. *Computational Science and Discovery*, 8(1), 014008, 2015.

[5] Chang, C.-C.; Lin, C.-J. (2011). LIBSVM: A library for support vector machines, *ACM Transactions on Intelligent Systems and Technology 2*, 2(3), 27:1–27:27, 2011. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[6] Chollet, F., et al. (2015). Keras. `https://keras.io`, 2015.

[7] Claesen, M.; Simm, J.; Popovic, D.; Moreau, Y.; Moor, B. D. (2014). Easy Hyperparameter Search Using Optunity, *CoRR abs/1412.1114*, 2014.

[8] Domhan, T.; Springenberg, J. T.; Hutter, F. (2015). Speeding up automatic hyperparameter optimization of deep neural networks by extrapolation of learning curves, In *Proceedings of the 24th International Conference on Artificial Intelligence*, IJCAI'15, AAAI Press, 3460–3468, 2015.

[9] Florea, A. C.; Andonie, R. (2018). A Dynamic Early Stopping Criterion for Random Search in SVM Hyperparameter Optimization, In *14th IFIP International Conference on Artificial Intelligence Applications and Innovations (AIAI)* (Rhodes, Greece, May 2018), L. Iliadis, I. Maglogiannis, and V. Plagianakos, Eds., vol. AICT-519 of *Artificial Intelligence Applications and Innovations*, Springer International Publishing, Part 3: Support Vector Machines, 168–180, 2018.

[10] Griewank, A. (1981). Generalized decent for global optimization, *Journal of Optimization Theory and Applications*, 34, 11–39, 1981.

[11] Hall, M.; Frank, E.; Holmes, G.; Pfahringer, B.; Reutemann, P.; Witten, I. H. (2009). The WEKA data mining software: An update, *SIGKDD Explor. Newsl.*, 11(1), 10–18, 2009.

[12] Hansen, N.; Müller, S. D.; Koumoutsakos, P. (2003). Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (cma-es), *Evol. Comput.*, 11(1), 1–18, 2003.

[13] He, K.; Zhang, X.; Ren, S.; Sun, J. (2016). Deep residual learning for image recognition, *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 770–778, 2016.

[14] Hinton, G. E. (2012). A practical guide to training Restricted Boltzmann Machines, In *Neural Networks: Tricks of the Trade (2nd ed.)*, G. Montavon, G. B. Orr, and K.-R. MĂźller, Eds., vol. 7700 of *Lecture Notes in Computer Science*. Springer, 599–619, 2012.

[15] Hutter, F.; Hoos, H.; Leyton-Brown, K. (2014). An efficient approach for assessing hyperparameter importance, In *Proceedings of International Conference on Machine Learning 2014 (ICML 2014)*, 754-762, 2014.

[16] Kennedy, J.; Eberhart, R. C. (1995). Particle swarm optimization, In *Proceedings of the IEEE International Conference on Neural Networks*, 1942–1948, 1995.

[17] Kirkpatrick, S. (1984). Optimization by simulated annealing: Quantitative studies, *Journal of Statistical Physics*, 34(5), 975–986, 1984.

[18] Kotthoff, L.; Thornton, C.; Hoos, H. H.; Hutter, F.; Leyton-Brown, K. (2017). Auto-WEKA 2.0: Automatic model selection and hyperparameter optimization in WEKA. *Journal of Machine Learning Research*, 18(25), 1–5, 2017.

[19] Krizhevsky, A.; Sutskever, I.; Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks, In *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1* (USA, 2012), NIPS'12, Curran Associates Inc., 1097–1105, 2012.

[20] LeCun, Y.; Bottou, L.; Orr, G.; MĂźller, K. (2012). *Efficient Backprop*, vol. 7700 LECTURE NO of *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. Springer, 9–48, 2012.

[21] Lemley, J.: Jagodzinski, F.; Andonie, R. (2016). Big holes in big data: A Monte Carlo algorithm for detecting large hyper-rectangles in high dimensional data, In *2016 IEEE 40th Annual Computer Software and Applications Conference (COMPSAC)*, 1, 563–571, 2016.

[22] Li, L.; Jamieson, K. G.; DeSalvo, G.; Rostamizadeh, A.; Talwalkar, A. (2016). Efficient hyperparameter optimization and infinitely many armed bandits. *CoRR abs/1603.06560*, 2016.

[23] Nair, V.; Hinton, G. E. (2010). Rectified linear units improve restricted boltzmann machines, In *Proceedings of the 27th International Conference on International Conference on Machine Learning* (USA, 2010), ICML'10, Omnipress, 807–814, 2010.

[24] Nelder, J. A.; Mead, R. (1965). A Simplex Method for Function Minimization, *Computer Journal*, 7, 308–313, 1965.

[25] Patterson, J.; Gibson, A. (2017). *Deep Learning: A Practitioner's Approach*, 1st ed. O'Reilly Media, Inc., 2017.

[26] Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; Vanderplas, J.; Passos, A.; Cournapeau, D.; Brucher, M.; Perrot, M.; Duchesnay, E. (2011). Scikit-learn: Machine learning in Python, *Journal of Machine Learning Research*, 12, 2825–2830, 2011.

[27] Shahriari, B.; Swersky, K.; Wang, Z.; Adams, R. P.; de Freitas, N. (2016). Taking the human out of the loop: A review of bayesian optimization. *Proceedings of the IEEE*, 104, 148–175, 2016.

[28] Smusz, S.; Czarnecki, W. M.; Warszycki, D.; Bojarski, A. J. (2015). Exploiting uncertainty measures in compounds activity prediction using support vector machines, *Bioorganic & medicinal chemistry letters*, 25(1), 100–105, 2015.

[29] Snoek, J.; Larochelle, H.; Adams, R. P. (2012). Practical bayesian optimization of machine learning algorithms, In *Advances in Neural Information Processing Systems 25*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2951–2959, 2012.

[30] Sobol, I. (1976). Uniformly distributed sequences with an additional uniform property, *USSR Computational Mathematics and Mathematical Physics*, 16(5), 236 – 242, 1976.

[31] Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. (2015). Going deeper with convolutions, In *Computer Vision and Pattern Recognition (CVPR)*, 2015.

[32] Thornton, C.; Hutter, F.; Hoos, H. H.; Leyton-Brown, K. (2013). Auto-WEKA: Combined selection and hyperparameter optimization of classification algorithms, In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (New York, NY, USA, 2013), KDD '13, ACM, 847–855, 2013.

[33] Zeiler, M. D.; Fergus, R. (2014). Visualizing and understanding convolutional networks. In *Computer Vision – ECCV 2014* (Cham, 2014), D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds., Springer International Publishing, 818–833, 2014.

[34] Zoph, B.; Le, Q. V. (2016). Neural architecture search with reinforcement learning, *CoRR abs/1611.01578* (2016).

[35] Zoph, B.; Vasudevan, V.; Shlens, J.; Le, Q. V. (2017). Learning transferable architectures for scalable image recognition. *CoRR abs/1707.07012*, 2017.

[36] [Online]. Bigml; BigML, Inc. `https://bigml.com/` Accessed: 2019-01-10.

[37] [Online]. Cifar 10; Krizhevsky, A., Nair, V., and Hinton, G. `https://www.cs.toronto.edu/~kriz/cifar.html` Accessed: 2019-01-10.

[38] [Online]. Google HyperTune, Google. `https://cloud.google.com/ml-engine/docs/tensorflow/using-hyperparameter-tuning` Accessed: 2019-01-10.

[39] [Online]. Optunity, `http://optunity.readthedocs.io/en/latest/`. Accessed: 2019-01-10.

[40] [Online]. SigOpt, SigOpt. `https://sigopt.com/` Sigopt. Accessed: 2019-01-10.

INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL
ISSN 1841-9836, e-ISSN 1841-9844, 14(2), 170-182, April 2019.

Heterogeneous Data Clustering Considering Multiple User-provided Constraints

Yue Huang*
School of Information Science
Beijing Language and Culture University, Beijing, China
*Corresponding author: huang.yuet@blcu.edu.cn

Abstract: Clustering on heterogeneous networks which consist of multi-typed objects and links has proved to be a useful technique in many scenarios. Although numerous clustering methods have achieved remarkable success, current clustering methods for heterogeneous networks tend to consider only internal information of the dataset. In order to utilize background domain knowledge, we propose a general framework for clustering heterogeneous data considering multiple user-provided constrains. Specifically, we summarize that three types of manual constraints on the object can be used to guide the clustering process. Then we propose the User-HeteClus algorithm to solve the key issues in the case of star-structure heterogeneous data, which incorporating the user constraint into similarity measurement between central objects. Experiments on a real-world dataset show the effectiveness of the proposed algorithm.
Keywords: clustering, heterogeneous networks, relational data, multi-typed objects, user constraints.

1 Introduction

With the advent of "big data", data mining has become a widely accepted tool for data analysis and various data mining-related research appeared [4,5,25,26]. Among all the techniques of data mining, clustering presents an effective way of exploring data, especially scenarios with no available labelled data. Compared with homogeneous networks, heterogeneous information networks [20] which consist of different types of objects and links can be found in many actual scenarios. Clustering on heterogeneous data has become a key emerging challenge during the recent twenty years [21, 22, 24].

In earlier study of clustering, we only have to deal with objects of the same type and numerous methods have been proposed [10]. Later, clustering on data with two different types of objects emerged, such as two-way clustering [9], co-clustering [1, 3, 6, 7], and bi-clustering [15]. Recently, more attention has been paid to multi-way clustering [2], also called high-order co-clustering [8]. Generally, in the datasets with real heterogeneity [13], the object type that contains less number of distinct values among heterogeneous data are called central type or target type, while the other types are called attribute types [19]. Whereas, most existing research focus on the star-structure of heterogeneous networks [11, 14, 18, 20], where links only exist between objects of the central type and objects of the attribute types, representing many scenarios in the real world. Besides, several methods have also been put forward for other types of heterogeneous networks [12] or arbitrary heterogeneous networks [16, 17].

In some scenarios, it is impossible to perform effective cluster analysis without taking advantage of the external information of the dataset, such as domain knowledge provided by users. Taking document analysis as an example, for a heterogeneous dataset containing tens of thousands of documents, thousands of words, and several document clusters, if the information of cluster assignment of one hundred pairs of documents is provided, then the clustering efficiency

on the documents can be improved. To solve the issue of clustering heterogeneous data considering user-provided constraints, we firstly analyzed that there are three types of constraint information provided by the user to help the clustering process of the objects. Then we propose a complete general analysis framework, based on which we propose corresponding solutions to solve the key issues in the case of star-structure heterogeneous data.

The rest of this paper is organized as follows. Section 2 defines the research scope of the problem and analyzes the hypotheses. Section 3 first proposes a general framework for solving the heterogeneous data clustering analysis that takes into account user-provided constraints, then it analyzes and resolves the key issues with the proposed algorithm UserHeteClus for heterogeneous data clustering considering multiple user-provided constraints. Section 4 gives the experimental results and analysis. Conclusion is given in Section 5.

2 Problem definition and hypothesis

Similar to previous study on semi-supervised clustering [23], it is easy to find that in heterogeneous data, user-provided constraints can be of three types: (1) the user labels which central objects should belong to (or do not belong to) the same cluster; (2) the user indicates which central objects with which attribute values must belong to (or not belong to) the same cluster; (3) the user indicates which attribute values actually correspond to the same (or different) meanings.

When the user can determine the cluster attribution of the central object by providing the values of the central object for one or some attribute objects, it indicates that the user provides a decisive attribute object. In this case, assigning the object cluster under such a constraint condition is similar to classifying central objects, requiring the user to provide very precise knowledge, so the condition is clearly too high to meet. According to previous studies and applications of semi-supervised clustering, what the user generally provides is information about the relationships between objects, but in practical applications, the type of constraint that the user may provide is not limited to this, so we have the following hypotheses:

Hypothesis 1: In the problem of heterogeneous data clustering considering user-provided constraints, the constraints may be on the central object or on the attribute object.

Hypothesis 2: In the problem of heterogeneous data clustering considering user-provided constraints, it is assumed that the heterogeneous dataset has a star structure.

3 Key issues and steps of the UserHeteClus

In this section, we propose the UserHeteClus algorithm for solving the key issues in the procedure of clustering star-structured heterogeneous data.

3.1 Framework for clustering heterogeneous data considering user constraints

The information available in the analysis of heterogeneous data with a star structure considering user-provided constraints includes internal information (object type information, object attribute information, and object relation information) and external information.

In the clustering of heterogeneous data with a star structure considering user-provided constraints, on the one hand, the composition of the dataset itself is very complex; on the other hand, it is necessary to integrate external information. Therefore, comprehensively, in the semi-supervised clustering of heterogeneous data considering user-provided constraints, we still need to follow the principle of "paying more attention to relations and less to attributes" [12], and it is necessary to separate the attribute similarity computation from object clustering.

In addition, relative to unsupervised clustering, semi-supervised clustering still has a small amount of useful sample information. Therefore, how to effectively utilize the domain knowledge contained in the labeled samples to guide the clustering process in the clustering algorithm is a key issue that distinguishes the semi-supervised clustering algorithm from the unsupervised clustering algorithm. In the traditional general framework for attribute-based clustering studies, the measurement of object similarity or dissimilarity (or distance) is the primary issue to be addressed, but in practice, the constraint information provided by the user is mostly at the object level rather than the attribute level, so it is very difficult to integrate user-provided constraints into the measurement process of similarity or dissimilarity. Therefore, except for the case in which the values of the user-provided attributes are identical, other user-provided constraint information should be used in the clustering process of the central object. Therefore, regarding this problem, the measurement of central object similarity and the process of central object clustering are two secondary key issues that need to be solved.

Based on the above discussions, we conclude that a complete procedure of heterogeneous data clustering considering user-provided constraints include four steps: (1)The presentation of user-provided constraints; (2) The measurement of the central object similarity; (3) The semi-supervised clustering of central objects considering user-provided constraints; (4) The clustering of attribute objects.

3.2 Presentation of user-provided constraints

At present, it is agreed in semi-supervised clustering studies that it is necessary to impose certain constraints on clustering results, i.e., the descriptions of the relation constraints between data objects can be categorized into two categories of must-link constraints and cannot-link constraints, with some respective properties. However, according to the above analyses, user-provided constraints are not limited to the above mentioned user constraints on object relations and may also include decisive attribute constraints and attribute value equality constraints.

Definition 1. User constraint on object relation: For the given heterogeneous data of $D=(C,A)$, assume that the user-provided constraint is on the central objects of $C = \{C_i | i = 1, 2, ..., n\}$ and that the central objects cluster as the set of *Clus* that contains N central object clusters, described as $C.Clus=\{Clus_1, Clus_2, ..., Clus_N\}$. Suppose that there are two central objects, i.e., C_s and C_t $(s \neq t)$. The must-link constraint and the cannot-link constraint of the user-provided constraint on object relation can be specifically described as follows:

- If C_s and C_t should be in the same cluster, then $Must-link(C_s, C_t) = True$, which can be specifically described as $C_s \in Clus_i$, $C_t \in Clus_j$, $i = j$;
- If C_s and C_t should not be in the same cluster, then $Cannot-link(C_s, C_t) = True$, which can be specifically described as $C_s \in Clus_i$, $C_t \in Clus_j$, $i \neq j$.

Therefore, the must-link constraint and the cannot-link constraint are both Boolean functions and have the following properties:

Remark 1. The must-link constraint and the cannot-link constraint of the user's two types of constraints have symmetry, and for $C_s, C_t \in C$:

- *Must-link*$(C_s, C_t) \Leftrightarrow$ *Must-link*(C_t, C_s);
- *Cannot-link*$(C_s, C_t) \Leftrightarrow$ *Cannot-link*(C_t, C_s).

Remark 2. The must-link constraint and the cannot-link constraint of the user's two types of constraints have limited transitivity, and for $C_r, C_s, C_t \in C$:

- *Must-link*(C_r,C_s)&&*Must-link*(C_s,C_t)⇒*Must-link*(C_r,C_t);
- *Must-link*(C_r,C_s)&&*Cannot-link*(C_s,C_t)⇒*Cannot-link*(C_r,C_t).

Definition 2. User Constraint on Decisive Attribute: For the given heterogeneous data of D=(C,A), assume that the user-provided constraint is on the central objects of $C = \{C_i|i = 1, 2, ..., n\}$ and that the central objects eventually cluster as the set of *Clus* that contains N central object clusters, described as $C.Clus$=$\{Clus_1, Clus_2, ..., Clus_N\}$. Suppose that there are two central objects, i.e., C_s and C_t ($s \neq t$), and that the cluster assignments of C_s and C_t are represented by $C_s.Clus$ and $C_t.Clus$, respectively. Then, a user-provided constraint on decisive attributes can be described as $C_s.\ A_{kp}$=$C_t.A_{kp} \Leftrightarrow C_s.Clus$=$C_t.Clus$, in which A_k is denoted as the decisive attribute.

Definition 3. User Constraint on Attribute Value Equality: For the given heterogeneous data of D=(C,A), assume that the user-provided constraint is on the central objects of $C = \{C_i|i = 1, 2, ..., n\}$ and that there are two central objects, i.e., C_s and C_t (s≠t), the user-provided constraint on attribute value equality can be described as $A_{kp} = A_{kq}$ ($p \neq q$).

3.3 Measurement of central object similarity

The measurement of central object similarity in heterogeneous data with a star structure adopts the following ideas: first, if the user provides a constraint on the attribute value equality, then it is used to get the heterogeneous data with unique identifiers; otherwise, this step is skipped. In a practical problem, the user may not be able to provide all the above-described three types of constraints. Then, when the central object similarity is measured pairwise, the similarity between the two central objects is represented using linear combinations of the two objects on each of attribute objects, in which the coefficient of each attribute object in the linear combination constitutes the contribution coefficient vector, which can be adjusted according to the actual situation. Therefore, the following basic concepts are defined.

Definition 4. Star-structured Data with Unique Identifiers (IDs): Star-structured heterogeneous data with unique IDs containing n central objects and r ($r \geqslant 1$) attribute objects are described as D'=(C,A,ID,R) (D'=(C,A) for short). The specific meanings are as follows:

- C represents the collection of central objects, i.e., $C = \{C_i\}_{i=1}^{n}$, where C_i represents the ith central object.
- A represents the collection of attribute objects, i.e., $A = \{A_k\}_{k=1}^{r}$ (k∈{1,2,...,r}), where $A_k = \{A_{kp}\}_{p=1}^{n_{A_k}}$ represents the object collection of the kth attribute ($p \in \{1, 2, ..., n_k\}$), A_{kp} represents the pth attribute object of the kth attribute object ($p \in \{1, 2, ..., n_k\}$), and n_{A_k} represents the number of attribute objects included in A_k.
- ID represents the collection of all objects, i.e., $ID = C.ID \bigcup A.ID$. $C.ID$ represents the collection of central objects with unique IDs, and $C.ID = \{C_i.ID\}_{i=1}^{n}$, where $C_i.ID$ represents the unique ID of the ith central object (i∈{1,2,...,n}). $A.ID$ represents the collection of attribute objects, and $A.ID = \{A_k.ID\}_{k=1}^{r}$ (k∈{1,2,...,r}), in which $A_k.ID = \{A_{kp}.ID\}_{p=1}^{n_k}$ represents the collection of the objects with the kth attribute ($p \in \{1, 2, ..., n_k\}$), where A_{kp} represents the pth attribute object of the kth attribute object ($p \in \{1, 2, ..., n_k\}$) and n_k represents the number of attribute objects included in A_k.
- R represents the undirected relationship collection present in the dataset of D, i.e., $R = \{r_l\}_{l=1}^{n_R}$ ($l \in \{1, 2, ..., n_R\}$), and for any relationship that is $r_l =< r_l.one, r_l.theother >$,

it satisfies the following condition: $r_l.one \in C$ and $r_l.theother \in A$ or $r_l.one \in A$ and $r_l.theother \in C$.

Given that no relationship between central objects is available in the star-structured heterogeneous data, the measurement of central object similarity can only rely on attribute objects, so the similarity of central object in terms of the remaining attribute objects is defined as follows:

Definition 5. Similarity between Central Objects in terms of the kth Type of Attribute Object: For the given heterogeneous dataset of $D'=(C,A)$, the calculation formula of the similarity, $S_k(C_i, C_j)$, between two central objects C_i and C_j in terms of the kth type of attribute object is as follows:

$$S_k(C_i, C_j) = \frac{2 \times |C_i.A_k.ID \cap C_j.A_k.ID|}{|C_i.A_k.ID| + |C_j.A_k.ID|} \tag{1}$$

where $C_i.A_k.ID$ represents the collection of IDs of the kth type of attribute object correlated to C_i and $C_j.A_k.ID$ represents the collection of IDs of the kth type of attribute object correlated to C_j.

Obviously, the value range of $S_k(C_i, C_j)$ is [0, 1], and it is easy to prove that it meets the properties of similarity measurement.

The linear combinations of the similarities of the central object on each of various attributes are considered to be used to measure the similarity between the central objects, but the roles of different attribute objects also differ, so we provide the following definition:

Definition 6. Similarity between Central Objects: For the given heterogeneous dataset of $D'=(C,A)$, the similarity, $S(C_i, C_j)$, between two central objects C_i and C_j is the linear combinations of similarities between the two in terms of the kth type of attribute object, with the following calculation formula:

$$S(C_i, C_j) = \sum_{k=1}^{r} w_k \cdot S_k(C_i, C_j) \tag{2}$$

where w_k is called the contribution coefficient, representing the contribution of the kth type of attribute object to the judgment of whether C_i and C_j are similar and satisfying: (1) $\sum_{k=1}^{r} w_k = 1$ and (2) $w_k \geqslant 0, w_k \in R$, which jointly constitute the contributing coefficient vector $w = (w_1, w_2, ..., w_k, ..., w_r)$.

Obviously, the value range of $S(C_i, C_j)$ is [0, 1], and it is easy to prove that it meets the properties of similarity measurement. w_k is evaluated according to the actual situation.

Central object similarity measurement of the UserHeteClus

Input: Star-structured heterogeneous data with unique ID ($D'=(C,A)$) and contribution coefficient vector ($w = (w_1, w_2, ..., w_k, ..., w_r)$), in addition to the constraint of attribute value equality.

Output: The similarity matrix $SimMatrix(C)$ of the central objects of $C = \{C_i\}_{i=1}^{n}$.

Procedures:

Step 1: The central objects of $C = \{C_i\}_{i=1}^{n}$ are represented using their attribute objects.

Step 2: According to the user-provided attribute value equality constraint, the value IDs of attributes with equal value are represented by one of the IDs; if the user does not provide the attribute value equality constraint, then skip this step.

Step 3: The similarity between any two central objects on each of various attribute objects $S_k(C_i, C_j)$ is calculated according to Definition 5.

Step 4: The similarity between any two central objects $S(C_i, C_j)$ is calculated according to Definition 6 to obtain the central object similarity matrix $SimMatrix(C)$.

3.4 Semi-supervised clustering of central objects considering user-provided constraints

Definition 7. Similarity between Central Object Clusters: For the given heterogeneous dataset of $D'=(C,A)$ and several central object clusters $Clus$, the similarity between central object clusters $Clus_s$ and $Clus_t$, $S(Clus_s, Clus_t)$, the average of the similarities between central objects contained in one cluster and those contained in another cluster, has the following formula:

$$S(Clus_s, Clus_t) = \frac{\sum_{i=1}^{|Clus_s|} \sum_{j=1}^{|Clus_t|} S(C_i, C_j)}{|Clus_s| \times |Clus_t|} \tag{3}$$

where $S(C_i, C_j)$ represents the similarity between C_i and C_j.

Obviously, the value range of $S(Clus_s, Clus_t)$ is [0, 1], and it is easy to prove that it meets the three properties of similarity measurement.

For the above mentioned three forms of user-provided constraint, the constraint of attribute value equality is related to the ID of the object, so it needs to be addressed first; the user constraint of the object attribute can be converted into the user constraint of the object relation, which is related to the clustering process, and can be incorporated into the specific clustering process. Therefore, the rationale for the semi-supervised clustering of central objects considering multiple user-provided constraints is that first, if the user provides a constraint on decisive attributes, then it is converted into a user constraint on the object relation, and then, the pairwise central objects similarities are sorted in descending order. After completing the above preparatory steps, the actual clustering process is executed. First, each object is treated as a separate cluster, and the objects in the must-link set are first linked. Then, the two most similar central objects are judged in terms of whether they meet the condition for the linking and sequentially repeated; if the two are in the cannot-link collection, then proceed to the next pair of central objects. Otherwise, the two are judged in terms of whether they have already in the same cluster, and if they are, then proceed to judging the next pair of central objects. Otherwise, the similarity between the cluster in which two objects reside is calculated, and if the similarity threshold is met, then the two objects are linked; if it is not, then the two are not linked. The process is repeated until the objects are linked as one cluster or the number of clusters is reached.

Semi-supervised clustering of central objects considering user-provided constraints of the UserHeteClus

Input: The collection of central objects $C = \{C_i | i = 1, 2, ..., n\}$, the central object similarity matrix $SimMatrix(C)$, the must-link set, the cannot-link set, the constraint of decisive object, the constraint of attribute value equality, and central object similarity threshold λ.

Output: Central object clustering result $C.Clus$.

Procedures:

Step 1: If the user provides a constraint on a decisive attribute, then convert it into a user constraint on object relation.

Step 2: The central object similarities are sorted in descending order.

Step 3: Each central object is treated as a separate cluster.

Step 4: Search the must-link collection to link the clusters in which the objects with the must-link constraint reside.

Step 5: Sequentially, for the two central objects with the highest similarity:

Step 5.1: Determine whether they are in the cannot-link set; if they are, then they are not linked, and continue to determine the next pair of central objects.

Step 5.2: Determine whether the two belong to the same cluster: if they do, then continue to determine the next pair of central objects.

Step 5.3: Determine whether the central object pair formed by the central objects composed of the two clusters that the two are from is present in the cannot-link set; if it is, then they are not linked, and continue to determine the next pair of central objects.

Step 5.4: Calculate the similarity between the two clusters that the two central objects are from. If the similarity is greater than or equal to λ, then link the two; otherwise, do not link, and continue to determine the next pair of central objects.

Step 5.5: Repeat Steps 5.1-5.4 until the objects are linked as one cluster or reach the required number of clusters.

3.5 Clustering of attribute objects

In the process of heterogeneous data clustering considering user-provided constraints, central objects are the focus of heterogeneous data clustering; because it is insufficient for the original attribute object information to support the clustering of the central objects, the user-provided partially labeled data are introduced and integrated into the process of clustering the central objects.

Therefore, in this study, we believe that the clustering of attribute objects should be based on the clustering result of the central objects and adopt the approach of "voting" using the nearest cluster of the attribute object to classify the attribute object to the central object cluster with the most votes and then separate it from the type of attribute object cluster, which is used in our previous study [12].

4 Experimental analysis of the UserHeteClus

4.1 Experimental data preparation

To test the clustering effectiveness of the UserHeteClus on actual data, in this section, we used the China National Knowledge Infrastructure (CNKI) literature data source to extract nonredundant records (2467 entries, searched on March 19, 2014) from the CNKI dataset of the Donlinks School of Economics and Management (DSEM) of University of Science and Technology, Beijing, using the four textual segments of paper title, author, source, and keyword. For convenience of describing the clustering results of the algorithm, records (in a total of 168 entries) that have no reference and the Chinese Library Classification (CLC) containing T were used as the testing dataset of the UserHeteClus (part of which is presented in Table 8) to analyze the research fields of the papers that are related to computer applications by authors from DSEM.

The constraints for this dataset include the following:

(1) The constraints of attribute value equality

A. The keywords "steelmaking - concasting", "concasting", and "steelmaking and concasting", and they were involved in 9 records.

B. The keywords of "clustering", "cluster analysis", and "clustering algorithm", and they were involved in 17 records.

(2) The constraints on user relations

A. Must-link constraint. The papers titled "Chinese keyword extraction algorithm based on high-dimensional clustering techniques", "Pattern aggregation theory-based text feature dimensionality reduction method and its application in text classification", and "Text classification based on granular network generation rules" belonged to the same cluster and were involved in three records.

Table 1: CNKI experimental dataset for the UserHeteClus

Id	Title	Author	Source	Keywords
59	Overview and analysis of manufacturing execution system models	Li Tieke	Metallurgical automation	Manufacturing execution system, information system, system model, enterprise model
62	Research on cryptographic algorithm in data transmission of Internet of Things of RFID system	Wang Xiaoni, Wei Guiying	Journal of Beijing Information Science and Technology University (Natural Science Edition)	Internet of Things, radio frequency identification, cryptographic algorithm
137	Telecom customer segmentation methods and applications	Chen Fengjie	Technology and industry	Data mining, clustering, Clementine
223	Global neighborhood algorithm for Job Shop scheduling problem	Cui Jianshuang, Li Tieke	Computer integrated manufacturing system	Neighborhood structure, critical path, job shop scheduling, neighborhood switching, scheduling algorithm
259	Research on CURE algorithm of hierarchical clustering method	Wei Guiying, Zheng Xuanxuan	Technology and industry	CURE algorithm, hierarchical clustering, clustering
268	Hybrid vehicle routing problem based on improved fuzzy genetic algorithm	Zhang Qun, Yan Rui	Chinese Management Science	Vehicle routing problem, fuzzy genetic algorithm, multidistribution center
...	...	...	...	...
2456	Research on the problem of determining the number of rolling units	Chen Xiong, Pan Yongquan	Control and decision	Rolling unit, simulated annealing algorithm, random variable variance, scheduling
2473	Diagnostic theory of two fuzzy control charts	Chen Zhiqiang, Zhang Gongxu, Yan Zhilin	Journal of Beijing University of Science and Technology	Shewhart control chart, selection control chart, total quality, subquality, fuzzy judgment

B. Cannot-link constraint. The papers titled "Basic framework and method for China's overseas mining investment decision process" and "High temperature compressive strength of coke for blast furnace" did not belong to the same cluster and were involved in two records.

In this experiment, two constraints were provided, and in practice, the three types of user-provided constraints may all be provided, or only one type may be provided.

4.2 Analysis of experimental results

(1) Analysis of object clustering accuracy

The UserHeteClus ($w_{author} = 0.3, w_{venue} = 0.2, w_{term} = 0.5, \lambda = 0.01$) generated 22 clusters

on 168 paper objects of the experimental dataset, which included two major clusters and 13 clusters of isolated points (Table 2). The meaning of each cluster was explained by looking at the title and keywords of the paper, and the cluster of isolated points was explained based on the title of the paper.

Table 2: Clustering result on paper objects of the experimental dataset with UserHeteClus

No.	Cluster size	Objects in the cluster	Interpretation
1	72	"1191", “1005", “1109", “945", “982", ...	Manufacturing execution system, production scheduling algorithm, and other papers
2	63	“383", “284", “947", “1865", “436", ...	Data mining papers
3	7	“687", “1067", “1284", “468",...	Software project management research
4	3	“2284", “2294", “2253"	Calculate accounting indicators with Excel
5	2	“2437", “2436"	Management information system development
6	2	“2346", “1383"	Supply chain and ERP
7	2	“1902", “1045"	System design and implementation using ASP.NET, etc.
8	2	“1796", “1551"	Temperature of nanoporous vacuum insulation panel
9	2	“955", “325"	Lean improvement research
10	1	“881"	The basic framework and method of China’s overseas mining investment decision-making process
11	1	“2330"	EU textile environmental label and its comprehensive evaluation
12	1	“1178"	Control technology of pipeline steel inclusions
13	1	“2271"	Application of value engineering in the development of building material products
14	1	“1921"	Research on factors and methods of reservoir management post evaluation
15	1	“393"	Recovery of the thorium resources of the mine in Baotou and its research status for nuclear fuel
16	1	“395"	Elliptic curve encryption algorithm and case analysis
17	1	“917"	Combination forecast of technical maturity of patented products of industrial pulverized coal boiler based on TRIZ theory
18	1	“298"	Investigation on the construction of evaluation index system for energy efficiency reform of existing buildings
19	1	“618"	Internet and e-commerce and logistics
20	1	“1680"	Enterprise system semantic complexity based on isomorphic ontology structure
21	1	“1535"	Current situation and development strategy of UPS
22	1	“274"	High temperature compressive strength of coke for blast furnace

The analysis of the central object clustering results indicates the following:

A. The classification on the macroclusters via the UserHeteClus is clear, with high inter-cluster discriminability, and has identified two major directions of computer-related studies, i.e., "manufacturing execution system" and "data mining", in DSEM, which are in line with the actual meaning of the cluster, with ideal clustering effectiveness.

B. For each specific cluster, clear meanings of the cluster are present, indicating that the intracluster similarity between the objects generated by the UserHeteClus is high.

C. The clusters of isolated points identified by the UserHeteClus are different from the major clusters, and the difference between the isolated points is also large.

(2) Analysis of the effect of main parameters of the algorithm

The UserHeteClus has two sets of parameters: w and λ. Specifically, the parameter set of w is used to control the weight of each type of attribute object when calculating the similarity

between the central objects, and λ is the central object similarity threshold, which affects and controls the clustering process of central objects. In this study, we also examined the effect of λ on the clustering result.

Since different values of λ lead to different numbers of central object clusters, the relationship between different values of λ and the number of central object clusters was tested on the UserHeteClus ($w_{author} = 0.3, w_{venue} = 0.2, w_{term} = 0.5$) using the CNKI experimental dataset (Fig. 1). It can be observed that a monotonically increasing relationship was present between λ and the number of central object clusters (paper cluster); specifically, when $\lambda = 0$, the number of paper clusters was 1; as the value of λ gradually increased, the number of paper clusters also gradually increased; when $\lambda = 1.0$, the number of paper clusters reached 168, i.e., the number of paper objects. Understanding the relationship between λ and the number of central object clusters helps to determine the optimal value of λ in practical applications.

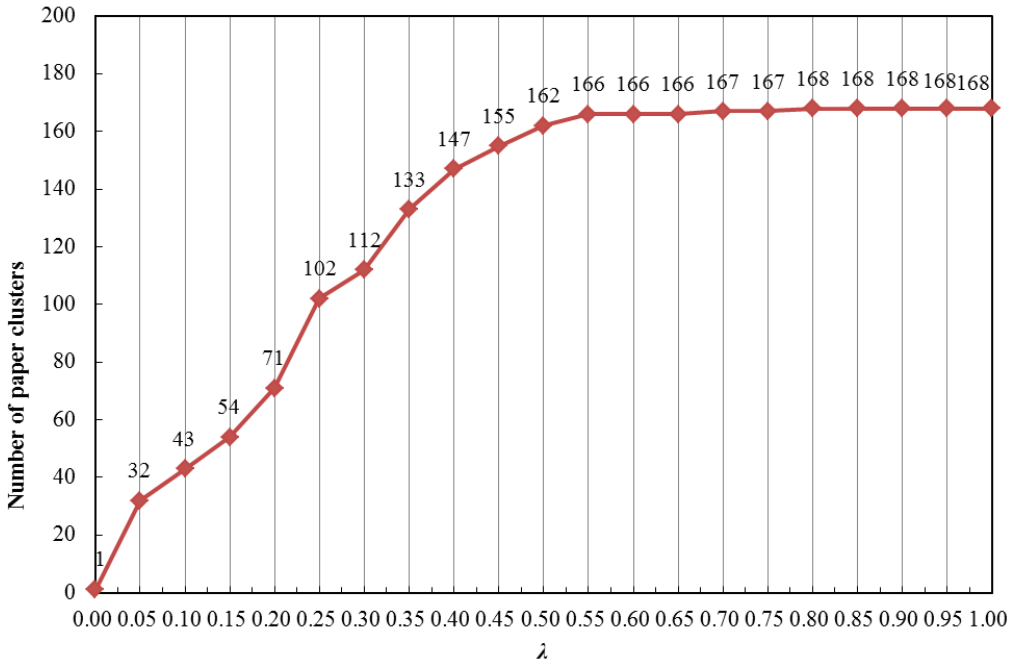


Figure 1: Relationship between the UserHeteClus parameter λ and the number of paper clusters

5 Conclusion

In this study, we investigated the issue of clustering heterogeneous data considering multiple user-provided constraints, in which the auxiliary external information about the object provided by the user is considered in addition to the information of the object itself. Firstly, we analyzed and we presented three types of constraint information that are provided by the user to help the clustering process of the objects, including constraints on user relations, constraints on user decisive attributes, and constraints on attribute value equality, which provide bases for future classification processes involving user-provided constraints in clustering algorithms. Secondly, we proposed a complete general analysis framework for clustering heterogeneous data considering user-provided constraints and then summarized the key issues for the case of star-structured heterogeneous data, including (1) the representation of user-provided constraints, (2) measurement of central object similarity, (3) semi-supervised clustering of central objects and (4) attribute

objects considering user-provided constraints. Lastly, we proposed using linear combinations of similarities of two central objects with all attribute objects to measure the similarity between the two. In addition, we defined the contribution coefficient to quantify the weight of various attribute objects on the central object similarity, based on which the similarities were sorted, enabling fast clustering of central objects under different types of user-provided constraints. For future work, we intend to investigate the issue of clustering heterogeneous data considering user constraints for arbitrary structure.

Funding

This work was partially supported by the Humanity and Social Science Youth Foundation of Ministry of Education of China (17YJCZH069), Science Foundation of Beijing Language and Culture University (supported by "The Fundamental Research Funds for the Central Universities") (19YJ040001) and BLCU Youth Talent Development Program.

Bibliography

[1] Banerjee, A.; Dhillon, I.S.; Ghosh, J. Merugu S.; Modha, D.S. (2004). A generalized maximum entropy approach to Bregman co-clustering and matrix approximation, *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 509–514, 2004.

[2] Bekkerman, R.; El-Yaniv, R.; McCallum, A. (2005). Multi-way distributional clustering via pairwise interactions, *Proceedings of the 22nd International Conference on Machine Learning*, 41–48, 2005.

[3] Chen, Y.; Wang, L.; Dong, M. (2010); Non-negative matrix factorization for semisupervised heterogeneous data coclustering, *IEEE Transactions on Knowledge and Data Engineering* 22(10), 1459–1474, 2010.

[4] Dai, Y.; Wu, W.; Zhou, H.; Zhang, J; Ma, F. (2018). Numerical simulation and optimization of oil jet lubrication for rotorcraft meshing gears, *International Journal of Simulation Modelling*, 17(2), 318–326, 2018.

[5] Dai, Y.; Zhu, X.; Zhou, H.; Mao, Z.; Wu, W. (2018). Trajectory tracking control for seafloor tracked vehicle by adaptive neural-fuzzy inference system algorithm, *International Journal of Computers Communications & Control*, 13(4), 465–476, 2018.

[6] Dhillon, I.S. (2001). Co-clustering documents and words using bipartite spectral graph partitioning, *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 269–274, 2001.

[7] Dhillon, I.S.; Mallela, S.; Modha, D.S. (2003). Information-theoretic co-clustering, *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 89–98, 2003.

[8] Gao, B.; Liu, T.; Ma, W. (2006). Star-structured high-order heterogeneous data co-clustering based on consistent information theory, *Proceedings of the Sixth IEEE International Conference on Data Mining*, 880–884, 2006.

[9] Getz, G.; Levine, E.; Domany, E. (2000). Coupled two-way clustering analysis of gene microarray data, *Proceedings of the National Academy of Sciences*, 97(22), 12079-12084, 2000.

[10] Han, J.; Kamber, M.; Pei, J. (2012). *Data Mining: Concepts and Techniques (Third Edition)*, Morgan Kaufmann Publishers, 2012.

[11] Huang, Y. (2016). A three-phase algorithm for clustering multi-typed objects in star-structured heterogeneous data, *International Journal of Database Theory and Application*, 9(8), 107–118, 2016.

[12] Huang, Y. (2017). Clustering multi-typed objects in extended star-structured heterogeneous data, *Intelligent Data Analysis*, 21(2), 225–241, 2017.

[13] Huang, Y.; Gao, X. (2014). Clustering on heterogeneous networks, *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 4(3), 213–233, 2014.

[14] Ienco, D.; Robardet, C.; Pensa, R.G.; Meo, R. (2013). Parameter-less co-clustering for star-structured heterogeneous data, *Data Mining and Knowledge Discovery*, 26(2), 217–254, 2013.

[15] Long, B.; Zhang, Z.; Wu, X.; Yu, P.S. (2006). Spectral clustering for multi-type relational data, *Proceedings of the 23rd International Conference on Machine Learning*, 585–592, 2006.

[16] Mei, J.; Chen, L. (2012). A fuzzy approach for multitype relational data clustering, *IEEE Transactions on Fuzzy Systems*, 20(2), 358–371, 2012.

[17] Pio, G.; Serafino, F.; Malerba, D.; Ceci, M. (2018). Multi-type clustering and classification from heterogeneous networks, *Information Sciences*, 425, 107–126, 2018.

[18] Rege, M.; Yu, Q. (2008). Efficient mining of heterogeneous star-structured data, *International Journal of Software and Informatics*, 2(2), 141–161, 2008.

[19] Sun, Y.; Han, J.; Zhao, P.; Yin, Z.; Cheng, H.; Wu, T. (2009). RankClus: integrating clustering with ranking for heterogeneous information network analysis, *Proceedings of the 12nd International Conference on Extending Database Technology: Advances in Database Technology*, 565–576, 2009.

[20] Sun, Y.; Yu, Y.; Han, J. (2009). Ranking-based clustering of heterogeneous information networks with star network schema, *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 797–806, 2009.

[21] Tang, L.; Liu, H. (2009). Uncovering cross-dimension group structures in multi-dimensional networks, *Proceedings of SDM Workshop on Analysis of Dynamic Networks*, 677–685, 2009.

[22] Tang, L.; Liu, H.; Zhang, J. (2012). Identifying evolving groups in dynamic multimode networks, *IEEE Transactions on Knowledge and Data Engineering*, 24(1), 72–85, 2012.

[23] Wagstaff, K.; Cardie, C. (2000). Clustering with instance-level constraints, *Proceedings of the 17th International Conference on Machine Learning*, 1103–1110, 2000.

[24] Yin, X.; Han, J.; Yu, P.S. (2006). LinkClus: efficient clustering via heterogeneous semantic links, *Proceedings of the 32nd International Conference on Very Large Data Bases*, 427–438, 2006.

[25] Zhang, W.; Zhang, Z.; Chao, H.; Tseng, F. (2018). Kernel mixture model for probability density estimation in Bayesian classifiers, *Data Mining and Knowledge Discovery*, 32(3), 675–707, 2018.

[26] Zhang, W.; Zhang, Z.; Qi, D.; Liu, Y. (2014). Automatic crack detection and classification method for subway tunnel safety monitoring, *Sensors*, 14(10), 19307–19328, 2014.

INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL
ISSN 1841-9836, e-ISSN 1841-9844, 14(2), 183-198, April 2019.

EODC: An Energy Optimized Dynamic Clustering Protocol for Wireless Sensor Networks using PSO Approach

C. Jothikumar, R. Venkataraman

C. Jothikumar*
Department of Computer Science
SRM Institute of Science and Technology
Chennai, India
*Corresponding author: jothikumar.c@ktr.srmuniv.ac.in

Revathi Venkataraman
Department of Computer Science
SRM Institute of Science and Technology
Chennai, India
revathi.n@ktr.srmuniv.ac.in

Abstract: A Wireless Sensor Network comprises many small wireless nodes that helps to sense, gather, process and communicate. One of its primary concerns is to optimize the consumption of energy and extend the network lifespan. The sensor nodes can be clustered to increase its lifespan further and this can be accomplished by choosing cluster head for every cluster and by performing data fusion on the cluster head (CH). The proposed system is using an energy efficient hierarchical routing protocol named Energy Optimized Dynamic Clustering (EODC) for clustering large ad-hoc WSNs and direct the data to reach the sink. The collected data is received by the sink from the set of cluster heads after every round. The cluster head is selected using the Particle Swarm Optimization (PSO) approach; the allocation of cluster members is based on the Manhattan distance. The metrics used to find the fitness function are location, link quality, energy of the active and inactive nodes. The system employs the shortest path approach to communicate between the cluster heads till it reaches the base station. By this, the energy efficiency and network lifetime have been increased. The analysis and outcomes show that the EODC was found to outperform the existing protocol which compares with this algorithm.
Keywords: Clustering, Wireless Sensor Network (WSN), energy efficiency, routing, particle swarm optimization (PSO).

1 Introduction

In recent times, Wireless Sensor Networks (WSN) are procuring research attention. The nodes communicate with their neighbors to transmit the data till it reaches the destination [2, 3]. The potential utilization of sensor networks is in various fields such as target monitoring, environment tracking, healthcare and industries. A WSN generally consists of sensor nodes, used for sensing, processing and communicating to reach the sink. These small sized sensor nodes make it easier to collect the data about the environment. A WSN has various characteristics, including less battery life, power constraints, coping with node failures, and dynamic network topology, etc.

Flooding is a flat based routing protocol which allows to flood the sensed data in the network through the neighbor nodes [8–10, 12]. Since the data are flooded throughout the network, the system drains due to duplication of data, overlapping and resource blindness. In direct diffusion the sink initiates the request to the source nodes. The source node saves the path of the sink, once the message is received. The routing consists of interest propagation and gradient setup. Interest propagation refers to the query from the sink where the gradient setup transmits the data

to reach the sink. An adaptive routing protocol called as the Sensor Protocol for Information via Navigation (SPIN) assigns high-level name to describe the data. The SPIN performs better by saving energy than the flooding approach. LEACH makes use a stochastic algorithm for selection of CH. The data gathered from the individual cluster member is fused by the CH and forwarded to reach the sink through single-hop communication. To avoid interference between the clusters while transmitting the data, a direct sequence spread spectrum is used. The CH is randomly chosen from the available nodes in the respective cluster to ensure that all the nodes should become a cluster head once. This algorithm can hence reduce the volume of data forwarded directly to a base station (BS). These cluster heads are unevenly distributed in the network. If they happen to be selected randomly only from a dense area, this may lead to increased energy consumption during data transmission. Since the CHs are elected based on probability, it may lead to overhead and increase in the network load and it is not scalable. The level of energy consumed is a prime concern in draining the network. In order to improve the efficiency, we developed an Energy Optimized Dynamic Clustering (EODC) protocol that uses location, link quality, energy of active and inactive nodes to find the fitness function for selecting the CH. The proposed system reduces the energy consumption by selecting the optimized CH and routes the packet through the near optimal path in the network.

The paper is structured as follows. The related work of the system is described in section 2. The Section 3 describes the EODC protocol. The results of the simulation and discussion are shown in section 4. At last, in section 5, the conclusion segment is covered.

2 Related work

The design of energy efficient routing protocol is such that it improves the performance and network lifespan. In this segment five hierarchical routing protocols are briefly discussed.

In a wireless sensor network, Low-Energy Adaptive Clustering Hierarchy-Centralized (LEACH-C) is the effective clustering approach for data communication. For the first round, the clusters are created and the selection of CH is based on a stochastic algorithm [11]. The BS receives the message about energy as well as the location of nodes. The BS computes the CH as per the received data. The nodes which have the energy more than the average of that set of nodes are considered as CHs from the second round. The CHs fuse the data gathered from the individual cluster member and forward it towards the BS. For saving energy, each member uses the time division multiple access (TDMA) scheduling for transmitting data to reach the CH and once the time is out, the node goes to sleep. The CH must be active to receive the data continuously and further pass the collected data to the base station. The LEACH-C being a one-hop communication protocol helps in direct transmission from CHs to BS. The system performs better than the flat based routing approach by employing a clustering technique.

Inside the Base Station Controlled Dynamic Clustering Protocol (BCDCP), the clusters are formed with an even number of nodes to balance the energy level [16, 20, 26]. The energy of overall nodes is computed by the BS. The nodes having the energy above the average are chosen as the CH for the next round. The selections of the CH are uniformly placed throughout the sensor field to maximize the distance between the CH in each splitting step. The received data are transmitted to the required CH using TDMA scheduling. The CH aggregates the received data before forwarding it. The protocol performs the multi-hop routing technique among the CHs for data forwarding using the minimum spanning tree approach. A single CH communicates the information to the BS; the involved CH communicating the data will have high energy for transmission.

Inside the General Self-Organized Tree-Based Energy-Balanced Routing Protocol (GSTEB), the root node is assigned by the BS depending on the remaining energy and its associated ID

is broadcasted to every node [28, 29]. Similarly, the nodes in the sensor region, select their parents based on the energy of the node by coordinating with their neighbor and every child node transmits the data to their parents within the given TDMA slot. The parent node communicates with the neighbor parents in the form of a tree structure till the data reaches the root node. Here, the data is fused and forwarded by the root node to the BS. The root node is dynamic to balance the load in the communication region. The energy of the sensor nodes in each round is known by the BS and this message favors in creating the topology for the round coming next.

The Hybrid Hierarchical Clustering Approach (HHCA) is a clustering hierarchy approach that performs the operation of the dual clustering mechanism with distributed clustering and centralized grid [14, 23, 25, 27]. Distributed clustering is a clustering approach where the nodes are distributed randomly and divided into sub clusters and the cluster members pass the data to the cluster head in the sub cluster. The CH in the sub cluster converses the shared data to the centralized grid. The BS computes the fuzzy C- means approach for the centralized grid and broadcasts this message in the network. If the ID of the node matches with the broadcast ID that node will act as the centralized grid. Distributed clustering uses the LEACH approach for clustering and CH selection. The cluster members utilize the TDMA schedule to share the data to the sub cluster. The sub cluster (distributed clustering) transmits the data to the centralized cluster (grid) through CSMA or CA mechanism. The centralized grid forwards the data to the sink directly without correlation.

Heuristic Algorithm for Clustering Hierarchy (HACH) is a clustering technique that performs the operation of the clustering mechanism with inactive nodes by selecting the low energy nodes to be in the sleep state based on a stochastic process that should not affect the network coverage [5, 6, 13, 18, 21]. The maximum coverage effect is employed in the network to select the inactive nodes to maintain the coverage area. The sink computes the Euclidean distance between the nodes and to itself. The computations are based on the coordinates and energy values received by the sink from all the nodes. The CH selection was broadcast to every node present in the network. If the node ID and the ID received by the node are the same, then that node will be the CH. The election of the CH deals with a heuristic crossover approach that is spatially distributed in the detecting range, and the selected CH has high energy. As far as the proposed system is concerned, we consider LEACH-C, HHCA and the HACH protocol for comparison.

3 Energy Optimized Dynamic Clustering (EODC) protocol

3.1 Network model

The nodes are thrown randomly into a square region to observe the environment continuously. N represents the total number of nodes, where $N = \{n_1, n_2, ..., n_n\}$. Assumptions carried out in a sensing region are

- The BS is fixed at the top of the sensing region
- All nodes are stationary in the deployment region
- Being homogeneous, the nodes have the same energy
- The adjustment of transmission power is made according to the received signal strength indicator (RSSI) value of the nodes

3.2 Energy consumption model

The energy of the data transit includes transmission circuitry and the volume of the data being transmitted. Similarly, even the energy of reception includes the volume of data reception. The energy consumption equation is given as: The energy required to transmit a unit of data is,

$$E_{Tr}(m,d) = E_c(m) + E_{amp}(m,d)$$

$$\Rightarrow E_{Tr}(m,d) = mE_c + m\epsilon_{fs}d^2; d < d_0$$

$$\Rightarrow E_{Tr}(m,d) = mE_c + m\epsilon_{mp}d^4; d \geq d_0$$

The energy required to receive a unit of data is

$$E_{Rr}(l) = E_c(m) = mE_c$$

Where, E_{Tr} and E_{Rr} are the transmission and reception energy respectively, $E_c(m)$ is the energy consumption per bit of transceiver circuitry, m is the message transmitted, and d indicates the level of distance measured between the nodes. Depending on the transmission range, free space (d^2) or multipath (d^4) propagation is used.

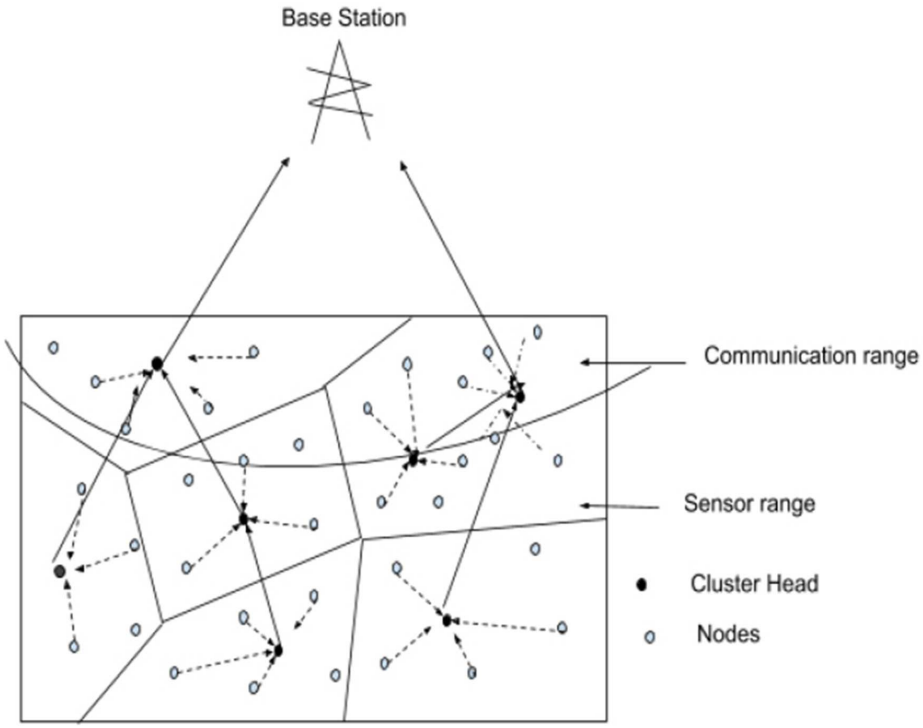


Figure 1: System Architecture of EODC

3.3 Operation of the EODC algorithm

In EODC, the nodes form the cluster using the PSO approach. The fitness function used is computed by the BS for the selection of CH and the cluster members (CM) are joined using the Manhattan distance. The purpose of the EODC is to reduce energy utilization in clustering

and data transmission [1, 4, 7, 15, 17, 19, 22, 24, 30]. The system employs the HACH approach for the selection of the inactive nodes in the clusters. The Stochastic Selection of Inactive Nodes (SSIN) protocol introduces the sleep scheduling mechanism to choose inactive cluster members. The coverage effect (C_e) is the effect of identifying the active nodes by putting some nodes to sleep according to the coverage. The number of active nodes present in the network is known which depends on the accumulated coverage effect (AC_e) of all the clusters. The received data are fused by the CH and forwarded to the nearest CH to form a multi- hop chain till the data reaches the BS. The CH in the communication region forwards the data to BS. A shortest path approach was employed to achieve the near optimal path to the BS.

The algorithm consists of two phases, namely:

1. Formation of Cluster and CH Selection using PSO;
2. Data Transmission and Forwarding.

Formation of cluster and CH selection using PSO

In the proposed system, the selection of CH adopting the PSO approach includes the location of the node, link quality between the nodes, and the energy of the active and inactive nodes. The CH should remain awake always for data reception and transmission. The fitness function of the CH is calculated using these metrics. F_{CH}, fitness function of CH is given as an optimization technique and is expressed as,

$$F_{CH} = \gamma \times LOC + (1-\gamma) \times LQ + (1-\gamma) \times E_n^{C_k} + (1-\gamma) \times E_m^{C_k} \tag{1}$$

Location(Loc) gives the identity of the nodes in the network. The computation gives the distance between the neighbor nodes, as well the distance staring from nodes until the CH of the cluster,C_k.

$$LOC = \min_{k=1,2,\ldots,K} \left\{ \sum_{n_i \epsilon C_k} \frac{d(n_i, CH_k)}{\mid C_k \mid} / \frac{d(n_i, BS)}{\mid C_k \mid} \right\} \tag{2}$$

Where, $d(n_i, CH_k)$ gives the distance from n_i to $CH_k, d(n_i, BS)$ represents the distance from n_i to BS, and $\mid C_k \mid$ is the length of the cluster

Link Quality (LQ) is the quality of the signal between nodes, ni and the CH in the cluster. The nodes in the CH receive the advertisement message. The strength of the signal is known by the nodes in message reception. The RSSI or ETx (Expected Transmission Count) is used to calculate the quality of the link, LQ. Here, ETx is a measure of transmission for the successful delivery of a packet from the sender to the receiver. The computation of paths along the links refers to ETT (expected transmission time) and is inversely proportional to link quality.

Therefore, $RSSI \alpha ETx \alpha (\frac{1}{ETT})$

$$LQ = \min_{k=1,2,\ldots,K} \left\{ \sum_{n_i \epsilon C_k} \frac{ETx_{(n_i, C_k)}}{\mid C_k \mid} \right\} \tag{3}$$

Where, $ETx_{(n_i, C_k)}$ is the maximum successful transmission of data from n_i to C_k, and $\mid C_k \mid$ is the length of the cluster.

The energy of the active nodes ($E_n(C_k)$) inside the network is the set of non-CH nodes which address the CH through TDMA scheduling and the current CH. The energy consumption of the

cluster head (ECH) includes the transmission energy of the advertisement message and TDMA schedule. The reception energy includes JOINT_REQ message and DATA message. The total energy dissipation of the CH is represented as,

$$E_{CH} = E_{Tr}(CH\_ADV, CH\_TDMA) + E_{Rr}(JOINT\_REQ\_CH, DATA\_CH)$$

Similarly, the energy dissipation of the non CH (active nodes) belonging to the cluster is represented as:

$$E_{non-CH} = E_{Tr}(JOINT\_REQ\_CH, DATA\_CH) + E_{Rr}(CH\_ADV, CH\_TDMA)$$

The nodes which communicate with the CH in the current round are called as active nodes. The energy requirement of the active nodes belonging to the network is given as the sum of all the optimum clusters.

$$E_n^{C_k} = \sum_{k=1, n_i \epsilon C_k}^{K} \frac{E(n_i^k - 1)}{E(CH_k)} \tag{4}$$

Where, $E(CH_k)$ is the active cluster heads energy.

The energy of the inactive nodes ($E_m^{C_k}$) is the set of nodes in the clusters which remain in sleep mode in the current round and on further iteration, these nodes will wake and another set of nodes become inactive.The energy of the inactive nodes is calculated as,

$$E_m^{C_k} = \sum_{k=1, m_j \epsilon C_k}^{k} \frac{E(m_j^k)}{minE(m_j^k)} \tag{5}$$

Where, $min(E_m^{C_k})$ is the minimum energy of the inactive nodes.

After computing the metrics, the fitness function that needs to minimize is given as

$$F_{CH} = 0.25 \times LOC + 0.25 \times LQ + 0.25 \times E_n^{C_k} + 0.25 \times E_m^{C_k} \tag{6}$$

Where, $\gamma = 0.25$

The fitness function is computed by the BS which broadcasts the message throughout the network. If the node ID matches with the ID of the node sent by the BS, then the node will act as a CH. The energy consumed by the CH includes the energy of data reception, fusion, and transit. The node will act as CH till the fitness function value is lesser than the threshold value. The energy of data communication includes the length of the data and energy required for the transmitter circuitry in addition to the energy required for amplification to forward the data through free space propagation, if $d < do$. The consumption of energy in the CH for data transmission is given as,

$$E_{CH} = mE_c(u_n^C - 1) + mE_{DA}(u_n^C) + mE_c + m\epsilon_{fs}d^2 \tag{7}$$

where u_n^C represents the number of unique nodes in each cluster.

Similarly, the energy of data reception includes the length of the data and energy required for receiver circuitry. The energy consumed by the non-CH for data transmission is given as,

$$E_{non-CH} = mE_c + m\epsilon_{fs}d^2 \tag{8}$$

The total consumption of energy for the single cluster includes the consumption of energy for the non-CH, of the cluster in intra cluster communication in addition to the energy consumption of the CH in the given cluster. The total energy consumed by the clusters is calculated as,

$$E_{clust} = (u_n^C - 1)E_{non-CH} + E_{CH} \tag{9}$$

Since, N represents the number of nodes and M indicates the sensor region, the selection of optimal clusters in the system is given using the equation,

$$K_{opt} = \frac{\sqrt{N}}{\sqrt{2n}}\sqrt{\frac{\epsilon_{fs}}{\epsilon_{mp}}} \times \frac{M}{d^2} \tag{10}$$

The energy consumed by the entire network is the product of consumption of energy for the single cluster and the optimal number of clusters, K_{opt} present in the sensing region. The total energy consumption by the network is given as,

Substituting (9) and (10) in (11)

$$E_{network} = K_{opt} \times E_{clust} \tag{11}$$

The assumption is being carried out as uniform deployment of nodes in the detecting region. The distance from the nodes to the CH is given as

$$E[d] = \int\int \sqrt{x - x_{CH}^2 + (y - y_{CH})^2}\rho(x,y)dxdy$$

$$E[d] = \int\int \sqrt{x^2 + y^2}\rho(x,y)dxdy$$

$$E[d] = \int\int r^2\rho(r,\theta)drd\theta \tag{12}$$

Here, $\rho(r,\theta)$ becomes constant, when the sensors are distributed uniformly in the environment.

PSO Approach: The optimization algorithm deals with the particles' velocity and position in a swarm. The movement of the particles in the swarm is referred to using the equation given. Movement of the particle to the next position from the current position is given as

$$p^{k+1} = p^k + v^{k+1} \tag{13}$$

The particle's velocity is given as

$$v^{k+1} = v^k + x_1 r_1(P_p^k - p^k) + x_2 r_2(S_p^k - p^k) \tag{14}$$

Where, P_p^k is the particle's best position, S_p^k is the swarm's best position, p is the particle position, v is the path direction, x_1, x_2 are the learning coefficients commonly set to 2.0, and r_1, r_2 are the random numbers with the range of (0,1).

Data transmission using shortest path approach

In this phase the data is transmitted from CH to CH till it reaches the base station. The CH in the communication region forwards the data to the sink since it needs high transmission power. A near optimal path is identified based on the route selection using the energy of the route. The energy of the route in the network is represented as

$$E(R_k) = \sum_{i=1}^{K} (E_i^{CH} - E_{i\_tx}^{CH}) \tag{15}$$

Where $E(R_k)$ is the energy of the route k, E_i^{CH} indicates the energy of the CH, and $E_{i\_tx}^{CH}$ denotes the transmission energy of the CH. The route selection is based on inter cluster communication till the data reaches the sink. The route which has the maximum energy to carry the data to the BS is taken as route assignment, R_A.

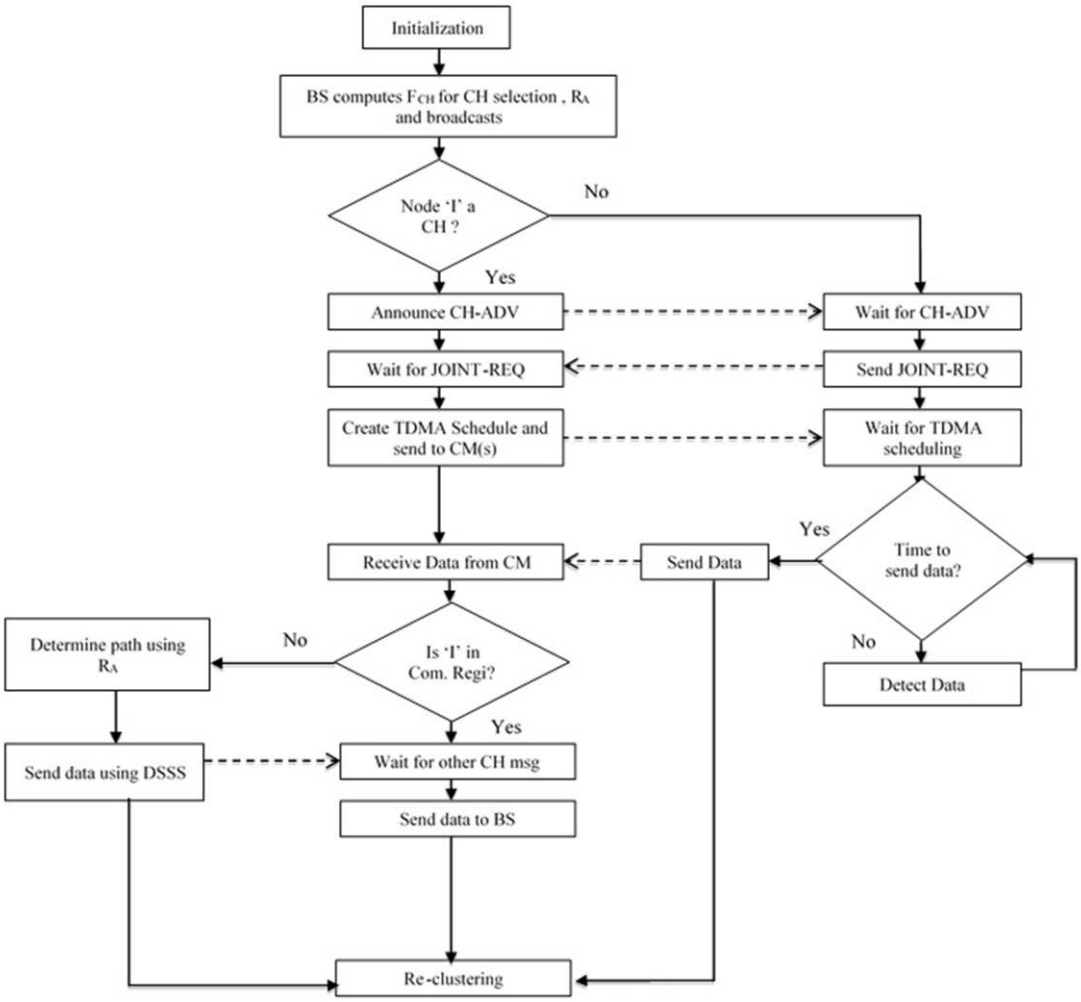


Figure 2: Flow chart of the cluster setup and data transmission

$$R_A = max\Big\{E(R_k); R_k \epsilon P\Big\} \tag{16}$$

where P is a set of all possible routes, k is a CH that follows the route reaching the sink, and R_A specifies the route assignment by condition. The cluster setup and data transmission flowchart is given in figure 2.

4 Simulation results

The simulation was done using the math-lab simulator. The EODC performance is compared with the HHCA, HACH and LEACH-C protocol. In our simulation work, about 500 sensor nodes are placed randomly in a 500m x 500m region. The initial energy of the node is calculated as 0.5J, the value of gamma is set to 0.25 and the size of the data was 500 bytes long. The simulation results are plotted in the graph and shown in figures 3 to 8. The values of the simulation parameters for the scheme are given below,

Table 1: Simulation Parameters

No.	Parameter	Specification
1	Senor Region	500 x 500 m^2
2	Number of sensors	500
3	Energy of each node	0.5J
4	Data size	500 byte
5	Ec	50 nJ/bit
6	$\epsilon_f s$	10 pJ/bit/m^2
7	ϵ_{mp}	0.0013 pJ/bit/m^4
8	γ	0.25
9	Minimum Threshold energy	$10^{-4}J$

The number of communication rounds increases when compared with other protocols such as LEACH-C, HHCA and HACH. The network lifespan is proportional to the number of communication rounds. The average energy dissipation at the end of 1000 rounds in EODC is 0.39nJ less than HHCA which gives 0.48nJ and in HACH, LEACH-C the energy of the nodes drains before 1000 rounds. Figure 3 illustrates the average energy dissipation for all the nodes. The EODC performs much better than the existing protocol.

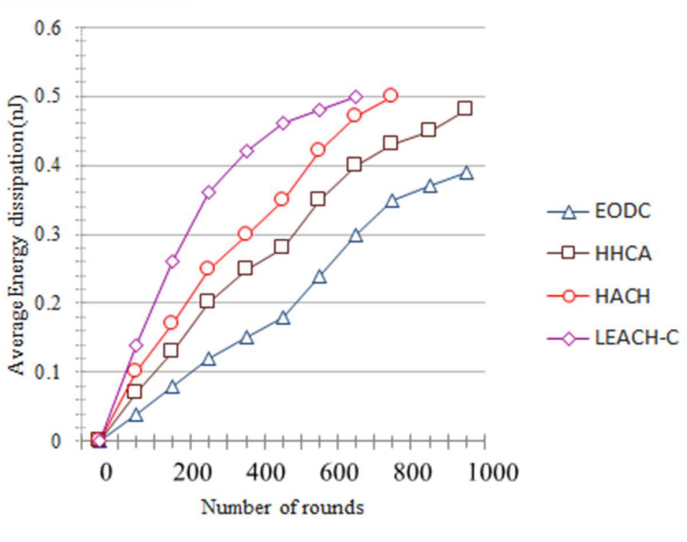


Figure 3: Comparison of average energy dissipation

In the proposed system, the number of live nodes in the network is high when compared with the existing protocol with respect to the number of rounds. At the end of 1000 rounds in EODC, the number of alive nodes is 95; whereas in HHCA, they are 25 and in HACH, LEACH-C, the nodes are dead before 1000 rounds. Figure 4 plots the number of live nodes in each iteration.

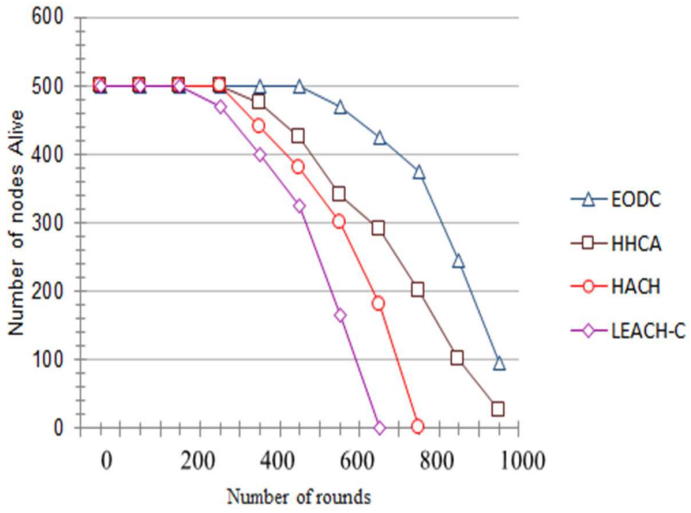


Figure 4: Number of alive nodes in each round

Figure 5 and Table 2 plot the clusters formed in the network in each round and the improvement achieved with EODC. In the proposed system, the count of clusters created at the end of 1000 rounds is 142, which is higher when compared with the existing protocols like HHCA, HACH and LEACH- C which give 117, 92, 79 respectively. Therefore, the improvements achieved in EODC over HHCA, HACH and LEACH- C are 21.37%, 54.35% and 79.75% respectively.

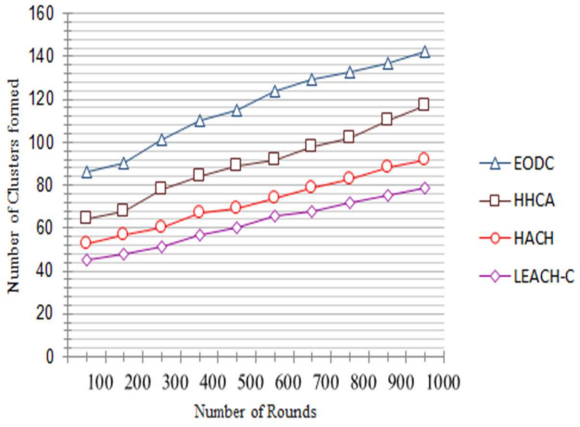


Figure 5: Clusters formed in each round

Table 2: Cluster Formation in each round of communication

Rounds	Number of clusters formed				Improvement achieved in % by EODC Over		
	EODC	HHCA	HACH	LEACH-C	HHCA	HACH	LEACH-C
100	86	64	53	45	34.38	62.26	91.11
200	90	68	57	48	32.35	57.89	87.50
300	101	78	60	51	29.49	68.33	98.04
400	110	84	67	57	30.95	64.18	92.98
500	115	89	69	60	29.21	66.67	91.67
600	124	92	74	66	34.78	67.57	87.88
700	129	98	79	68	31.63	63.29	89.71
800	133	102	83	72	30.39	60.24	84.72
900	137	110	88	75	24.55	55.68	82.67
1000	142	117	92	79	21.37	54.35	79.75

Figure 6 plots the energy dissipation of the CH with the given existing protocol. The energy consumed by the CH in EODC is 0.12nJ for 200 iterations which is less than those of the existing systems like HHCA, HACH and the LEACH-C protocol which give 0.18nJ, 0.21nJ and 0.29nJ respectively.

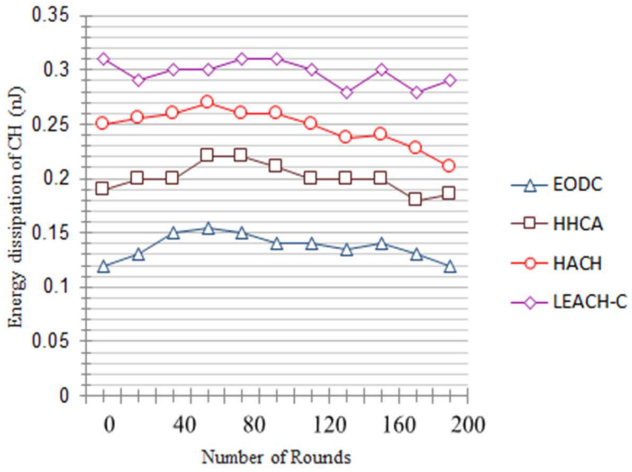


Figure 6: Energy dissipation of the CH under each round

Figure 7 and table 3 show the sum of residual energy during each round and the improvement achieved by EODC. In the proposed system, the residual energy of the sum of all nodes is evaluated and in comparison, the sum of network's residual energy is higher than those of the existing systems like LEACH-C, HHCA and the HACH protocol. The improvements achieved in EODC over HHCA, HACH and LEACH-C are 100%, 211% and 357.14% respectively.

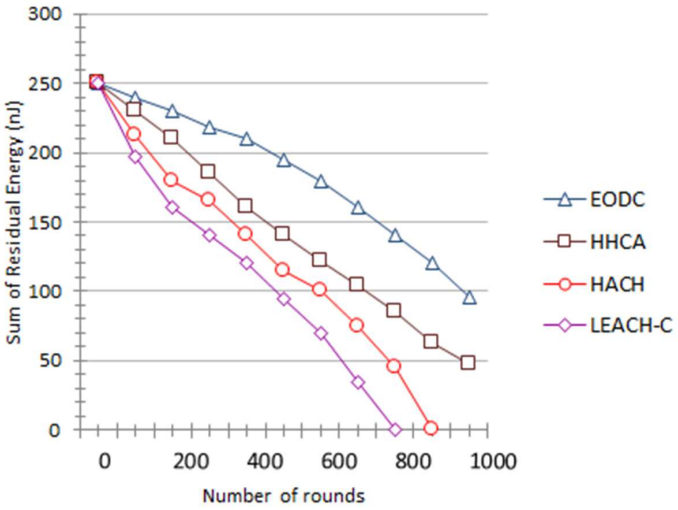


Figure 7: Sum of residual energy during each round

Table 3: Sum of Residual Energy during each round of communication

Rounds	Sum of Residual energy				Improvement achieved in % by EODC Over		
	EODC	HHCA	HACH	LEACH-C	HHCA	HACH	LEACH-C
0	250	250	250	250	0.00	0.00	0.00
100	240	230	213	197	4.35	12.68	21.83
200	230	210	180	160	9.52	27.78	43.75
300	218	185	165	140	17.84	32.12	55.71
400	210	160	140	120	31.25	50.00	75.00
500	195	140	115	95	39.29	69.57	105.26
600	180	122	101	70	47.54	78.22	157.14
700	160	104	75	35	53.85	113.33	357.14
800	140	85	45	0	64.71	211.00	-
900	120	63	0	0	90.48	-	-
1000	96	48	0	0	100.00	-	-

Figure 8 and table 4 plot the network lifetime comparison with LEACH-C, HHCA, and HACH with the EODC protocol. The lifetime gradually increases for the EODC protocol with the total number of nodes, which ranges from 250 to 500. As per the percentage of nodes dying in every round, we calculate the lifespan of the network. The improvements achieved in EODC over HHAC, HACH and LEACH-C are 12.85%, 44.77% and 82.18% respectively. The proposed protocol gives better performance in comparison

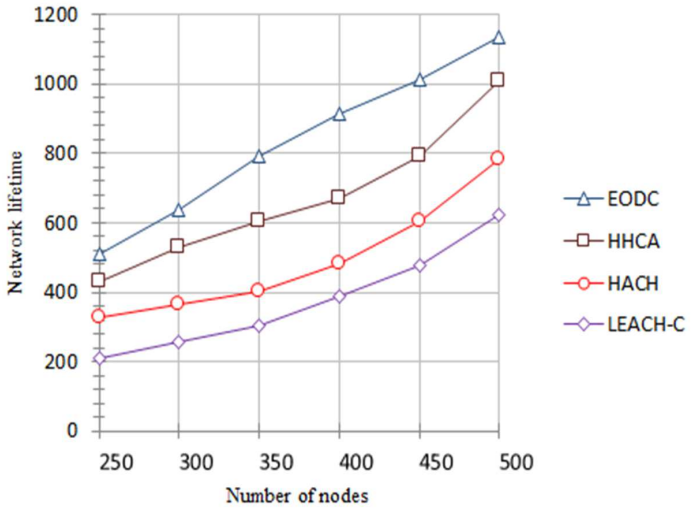


Figure 8: Lifetime Comparison of the Protocol

Table 4: Lifetime comparison of the protocols

No. of nodes	Lifetime Comparison				Improvement achieved in % by EODC Over		
	EODC	HHCA	HACH	LEACH-C	HHCA	HACH	LEACH-C
250	510	431	328	212	18.33	55.49	140.57
300	635	530	364	256	19.81	79.38	148.05
350	792	602	404	306	31.56	96.04	158.82
400	916	671	484	388	36.51	89.26	136.08
450	1014	793	604	477	27.87	67.88	112.58
500	1135	1006	784	623	12.85	44.77	82.18

From the projected graphs of the simulation results, it is concluded that EODC has a lower energy dissipation, increase in the generation of clusters and the sum of residual energy consumed is low when compared with the preceded LEACH-C, HHCA and HACH protocols, thus increase the lifespan of the nodes and efficiency of the data transmission across the plane. The performance of EODC is compared with the above mentioned protocols with respect to energy and network lifespan. Network lifespan is characterized as the count of rounds at the point until every one of the sensor nodes comes up short on energy. It should be observed that our proposed model still has some outstanding energy beyond 1100 rounds, whereas only slight energy is remaining beyond 1000 rounds with the HACH protocols.

5 Conclusion

Energy efficiency is an essential criterion for the lifespan of the sensor network. Clustering the nodes together is the potent way to promote the energy efficiency of the network. The EODC approach is a type of hierarchical routing protocol like LEACH-C, HHCA and HACH which perform clustering of the sensor nodes. LEACH, HACH and the HACH protocol randomly choose the CHs, which in turn, increases the re-clustering process; thus more energy is being used and the network lifespan gets affected. In EODC, the cluster heads are chosen based on the PSO approach. The metrics of the proposed system are compared with the PSO model. One of

the primary concerns is an effective CH selection which has been given in the proposed system. The simulation results report that in EODC, the number of clusters in the network is high in comparison. The inter cluster communication is performed using the shortest path algorithm in this scenario. The results show that by doing so the network lifetime is more in our approach when compared with the other hierarchical routing protocols. Our simulation results have proven that EODC produces better results than LEACH-C, HHCA and the HACH hierarchical routing protocol with reference to the total energy consumption and the lifetime of the network.

Acknowledgments

The authors wish to acknowledge the support extended by Department of Computer Science and Engineering , SRM Institute of Science and Technology, Chennai, India, to carry out this research work.

Author contributions. Conflict of interest

The authors contributed equally to this work. The authors declare no conflict of interest.

Bibliography

[1] Abdul Latiff, N.M.; Tsimenidis, C.C.; Sharif, B.S. (2007). Energy-Aware Clustering For Wireless Sensor Networks Using Particle Swarm Optimization, *18th Annual IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*, September 2007.

[2] Akyildiz, I.F.; Su, W.; Sankarasubramaniam, Y.; Cayirci, E. (2002). Wireless sensor networks: a survey, *Computer Networks*, 38(4), 393-422, 2002.

[3] Akkaya, A.; Younis, M. (2005). A Survey on Routing Protocols for Wireless Sensor Networks, *Elsevier Journal of Ad Hoc Networks*, 3(3), 325-349, 2005.

[4] Alphonse, P.J.A.; Sivaraj C.; Janakiraman T.N. (2017). An Energy-Efficient Layered Clustering Algorithm for Routing in Wireless Sensor Networks. *International Journal of Distributed Systems and Technologies*, 8(3), 43-66, July, 2017.

[5] Batra, P.K.; Kant, K.(2016). A clustering algorithm with reduced cluster head variations in LEACH protocol, *Int. J. Syst., Control Commun*, 7(4), 321-336, January 2016.

[6] Chang, C.-Y.; Chang, H.-R. (2008). Energy-aware node placement, topology control and MAC scheduling for wireless sensor networks, *Comput. Netw*, 52(11) , 2189-2204, August 2008.

[7] Elhabyan, R.S.; Yagoub, M.C.E. (2014). Energy Efficient Clustering Protocol for WSN using PSO, *IEEE Global Information Infrastructure and Networking Symposium*, September 2014.

[8] Hedetniemi, S.; Liestman, A. (1998). A Survey of Gossiping and Broadcasting in Communication Networks, *Networks*, 18(4), 319-349, 1998.

[9] Heinzelman, W.R.; Kulik, J.; Balakrishnan, H. (1999). Adaptive protocols for information dissemination in wireless sensor networks, *5th annual ACM/IEEE international conference on Mobile computing and networking (MobiCom '99). ACM* , DOI=http://dx.doi.org/10.1145/313451.313529

[10] Heinzelman, W.R.; Chandrakasan, A.; Balakrishnan H.(2000). Energy-efficient communication protocol for wireless microsensor networks, *Proceedings of the 33rd Annual Hawaii International Conference on System Sciences, Maui, HI, USA*, 2, 110, 2000.

[11] Heinzelman, W.B., Chandrakasan, A.P., Balakrishnan, H. (2002). An Application-Specific Protocol Architecture for Wireless Microsensor Networks, *IEEE Transaction of Wireless Communications*, 1(4), 660-670, 2002.

[12] Intanagonwiwat, C.; Govindan, R.; Estrin, D. (2000). Directed diffusion: a scalable and robust communication paradigm for sensor networks, *6th annual international conference on Mobile computing and networking (MobiCom '00). ACM*, DOI=http://dx.doi.org/10.1145/345910.345920

[13] Kandris, D.; Tsioumas, P.; Tzes, A.; Nikolakopoulos. G.; Dimitrios Vergados, D.(2009). Power Conservation Through Energy Efficient Routing in Wireless Sensor Networks, *Sensors*, 9(9), 7320-7342, 2009.

[14] Lee, J.S.; Kao, T.Y. (2016). An Improved Three-Layer Low-Energy Adaptive Clustering Hierarchy for Wireless Sensor Networks, *IEEE Internet of Things Journal*, 3(6), 951-958, 2016.

[15] Li, D.; Wen, X.. (2014). An improved PSO algorithm for distributed localization in wireless sensor networks, *International Conference on Software Intelligence Technologies and Applications & International Conference on Frontiers of Internet of Things 2014, Hsinchu*, 184-189, 2014.

[16] Lindsey, S.; Raghavendra, C. (2002). Data Gathering Algorithm in Sensor Networks Using Energy Metrics, *IEEE Transactions on Parallel and Distributed Systems*, 13(9), 924-935, 2002.

[17] Logambigai, R.; Kannan, A. (2018). Energy conservation routing algorithm for wireless sensor networks using hybrid optimisation approach, *International Journal of Communication Networks and Distributed Systems*, 20(3), 352-371, January, 2018.

[18] Lotf, J.; Bonab, M.; Khorsandi, S. (2006). A Novel Cluster-Based Routing Protocol with Extending Lifetime for Wireless Sensor Networks, *5th International Conference on Wireless and Optical Communications Networks, East Java Indonesia, Surabaya*, 1-5, 2006.

[19] Mao, J.; Wu, Z.; Wu, X. (2007). A TDMA scheduling scheme for many-to-one communications in wireless sensor networks, *Computer Communication*, 30(4), 863-872, 2007.

[20] Muruganathan, D.; Bhasin, R. (2005). A Centralized Energy Efficient Routing Protocol for Wireless Sensor Networks, *IEEE Communication Magazine*, 43(3), 8-13, 2005.

[21] Oladimejia, M.O.; Turkeya, M.; Dudleya, S. (2017). HACH: Heuristic Algorithm for Clustering Hierarchy Protocol in Wireless Sensor Network, *Applied Soft Computing*, 452-460, 2017.

[22] Rao, P. C.; Jana, Prasanta K.; Haider Banka. (2017). A particle swarm optimization based energy efficient cluster head selection algorithm for wireless sensor networks, *Wireless Networks*, 23(7), 2005-2020, 2017.

[23] Sangho Yi; Junyoung Heo; Yookun Cho; Jiman Hong.(2007). PEACH: Power-efficient and adaptive clustering hierarchy protocol for wireless sensor networks, *Comput. Commun*, 30, 2842-2852, 2007.

[24] Sarkar, A.; Senthil Murugan, T. (2019). Cluster head selection for energy efficient and delay-less routing in wireless sensor network. *Wireless Network*, 25(1), 303-320, 2019.

[25] Thein, M.C.M.; Thein, T. (2010). An energy efficient cluster head selection for wireless sensor networks, *Proceedings of the UKSim/AMSS 1st International Conference on Intelligent Systems, Modelling and Simulation*, 287-291, 2010.

[26] Wu, Y.; Fahmy, S.; Shroff, N.(2007). Energy Efficient Sleep/Wake Scheduling for Multi-Hop Sensor Networks: non-Convexity and Approximation Algorithm, *26th Annual IEEE Conference on Computer Communications, Anchorage, Alaska*, 1568-1576, 2007.

[27] Younis, O.; Fahmy, S. (2004). HEED: A Hybrid, Energy-Efficient Distributed Clustering Approach for Ad Hoc Sensor Networks, *IEEE Transactions on Mobile Computing*, 3(4), 366-379, 2004.

[28] Zhao, S.; Wu, J.; Jiezhang; LiefengLiu; kaiyun Tian.(2014). A General Self-Organized Tree-Based Energy-Balance Routing Protocol for Wireless Sensor Network, *IEEE Transactions on Nuclear Science*, 61(2), 732- 740, 2014.

[29] Zhang, D.; Li, G.; Zheng, K.; Ming, X.; Pan, Z.H.(2014). An Energy Balanced Routing Method Based on Forward-Aware Factor for Wireless Sensor Networks, *IEEE Transactions on Industrial Informatics*, 10(1), 766-773, 2014.

[30] Zhou, Y.; Wang, N.; Xiang, W. (2016). Clustering Hierarchy Protocol in Wireless Sensor Networks Using an Improved PSO Algorithm, *IEEE Access*, 5, 2241-2253, 2016.

INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL
ISSN 1841-9836, e-ISSN 1841-9844, 14(2), 199-211, April 2019.

Optimal Data File Allocation for All-to-All Comparison in Distributed System: A Case Study on Genetic Sequence Comparison

L.X. Li, J. Gao, R. Mu

Leixiao Li
1. College of Computer and Information Engineering,
Inner Mongolia Agricultural University, Hohhot 010018, China
2. College of Data Science and Application,
Inner Mongolia University of Technology, Hohhot 010080, China
3. Inner Mongolia Autonomous Region Engineering & Technology
Research Center of Big Data Based Software Service, Hohhot 010080, China
llxhappy@126.com

Jing Gao*
College of Computer and Information Engineering,
Inner Mongolia Agricultural University, Hohhot 010018, China
*Corresponding author: gaojing@imau.edu.cn

Ren Mu
State key laboratory,
Beijing Jiaotong University, BeiJing 100044, China
568387304@qq.com

Abstract: In order to solve the problem of unbalanced load of data files in large-scale data all-to-all comparison under distributed system environment, the differences of files themselves are fully considered. This paper aims to fully utilize the advantages of distributed system to enhance the file allocation of all-to-all comparison between the data files in a large dataset. For this purpose, the author formally described the all-to-all comparison problem, and constructed a data allocation model via mixed integer linear programming (MILP). Meanwhile, a data allocation algorithm was developed on the Matlab using the *intlinprog* function of branch-and-bound method. Finally, our model and algorithm were verified through several experiments. The results show that the proposed file allocation strategy can achieve the basic load balance of each node in the distributed system without exceeding the storage capacity of any node, and completely localize the data file. The research findings can be applied to such fields as bioinformatics, biometrics and data mining.
Keywords: distributed system, all-to-all comparison, mix integer linear programming (MILP), file allocation, load balancing.

1 Introduction

The comparison between two random data files in a dataset is commonplace in bioinformatics, biometrics and data mining [1]. However, the all-to-all comparison between data files in a large dataset require special large scale computations. Previous solutions to such a problem mainly fall into two categories: those grounded on centralized computing or those using distributed computing based on centralized storage [2]. The former requires access to supercomputer resources [3], while the latter is bottlenecked by limited storage capacity and task delay caused by waiting for data transmission [4].

The distributed computing based on distributed storage provides an efficient, reliable and scalable solution to large-scale computing problems like all-to-all comparison. By distributed

computing, a large-scale problem is decomposed into several small problems, which are then handled separately on each node in a distributed system [5]. Nevertheless, the performance of distributed computing depends heavily on data allocation, task decomposition and task scheduling, and might be dampened by the irrational data allocation, poor data locality and computing load imbalance in the distributed system [6].

There are two data allocation strategies for all-to-all comparison of data files in a large dataset via distributed computing based on distributed storage, namely, allocating all input files to each computing node, and allocating a number of copies of each input files randomly to the system node (i.e. the Hadoop framework data allocation policy) [7–9]. Hadoop framework can not guarantee the load balancing of the comparison task calculation of each node. Storing all input files on each node is a common practice in centralized computing solutions, but it wastes storage and network resources [10] For the latter strategy, the computing performance is poor due to the frequency data movement between the nodes [11]. After all, Hadoop, as a general distributed computing framework, is not designed specifically for all-to-all comparison [1, 12].

To solve the above problems, this paper establishes a model to optimize the data file allocation optimization in all-to-all comparison of data files in a large dataset, puts forward a data file allocation algorithm and a task scheduling strategy, and verifies the proposed methods via experiments. The experimental results show that our model and algorithm can achieve data localization and load balancing of comparative files under the storage constraints of distributed system nodes, thus giving full play to the advantages of distributed system.

The remainder of this paper is organized as follows: Section 2 describes the all-to-all comparison problem and constructs a data file allocation model; Section 3 designs the data allocation algorithm; Section 4 performs the experimental verification and discusses the results; Section 5 wraps up this paper with several conclusions.

2 Problem description and data file allocation model

2.1 Problem description

All-to-all comparison refers to the multiple contrasts of all data files in a dataset. This problem can be illustrated by the graph in Figure 1, where each vertex is a data file to be compared and each edge is a comparison task between two data files. If m is the number of data files to be compared, then the total number of comparison tasks is $m(m-1)/2$. Hence, the all-to-all comparison problem can be expressed as a graph with m vertices and $m(m-1)/2$ edges.

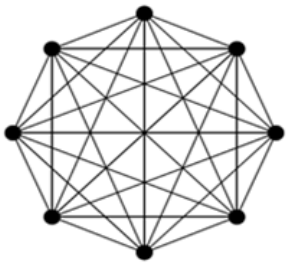


Figure 1: All-to-all comparison problem

In a distributed system with distributed storage, the all-to-all comparison of data files in a large dataset is implemented in two steps: First, all data files in the dataset are allocated to each computing node of the distributed system; then, the data files are compared in a pairwise manner.

Table 1: Symbol description

Number	Symbol	Symbolic description
1	m	Number of data files
2	n	Number of computing nodes in distributed system
3	s	M row 1 column matrix, representing the size of each file
4	$s_i, i = 1, 2, \ldots, m$	Size of i data file
5	u	N row 1 row matrix, representing the maximum storage limit of each node in a distributed system
6	$u_i, i = 1, 2, \ldots, n$	The storage capacity of computing nodesi
7	$w_{ij}, i = 1, 2, \ldots, m, j = 1, 2, \ldots, m$	The size of the task compares with file i and file j.
8	$c_{ij}, i = 1, 2, \ldots, m, j = 1, 2, \ldots, m$	Amount of calculation of the task compares with file i and file j.
9	$x_{kt}, k = 1, 2, \ldots, m(m-1)/2, t = 1, 2, \ldots, n$	whether or not allow the assignment of item K allocated to the T node
10	$W_{kt} = w_{ijt}$	Assign the size of item K to the T node (the file number corresponding to item K is i, j).
11	$C_{kt} = c_{ijt}$	The amount of computation assigned to item K of the T node (the file number corresponding to item K corresponds to i, j).
12	$taskno$	Task number
13	$av_$work	The average amount of tasks per computing node should be allocated in theory,
14	$deci$	Correspondence matrix between task and file
15	$result$	Task allocation result matrix
16	f	Objective function variable coefficient matrix
17	aeq	Coefficient matrix of equality constraint variables
18	beq	Equality constrained variable resource matrix
19	a	Inequality constraint coefficient matrix
20	b	Inequality constrained resource matrix
21	$intcon$	Integer variable subscript sequence number vector
22	LB	Lower limit of variable
23	UB	Upper limit of variable
24	$[X, Y]$	X is the best solution for obtaining the variable value. Y is the best solution.

Before allocating the data files, it is necessary to fully consider how the system performance is affected by node storage capacity, data transmission, network bandwidth, load balance and other factors. A desirable allocation strategy must satisfy the following conditions:(1) The data allocated to a node should not surpass the storage capacity of that node;(2) The two files to be compared on a node must be saved on that node to localize the data files for each comparison task;(3) The comparison tasks should be balanced among the computing nodes.

For simplicity, all the symbols used in this paper are listed in Table 1 below.

For better system speed and performance, our goal is to balance the comparison tasks among the computing nodes under the storage capacity of each node.

2.2 Data file allocation model

The above description shows that all-to-all comparison is a typical constrained optimization problem: maximizing or minimizing objective functions under multiple constraints. The most effective solution to constrained problem is linear programming (LP), which works well when the objective functions and constraints are all linear [13,14]. With a strong modelling ability, the LP is also a desirable way to tackle control and programming problems. The main idea of the LP is to find a control sequence that satisfies all constraints and minimizes the objective function [15].

In this paper, node storage is added to the constraints of the all-to-all comparison problem, and load balancing is treated as the objective function. All constraints were expressed as an equality or inequality. As mentioned before, our problem involves m data files and n nodes in a distributed system with distributed storage. For multiple comparisons of all data files, the m data files should be distributed rationally to those nodes to fulfill load balancing, data localization and node storage constraint.

As shown in Figure 1, the total number of comparison tasks of these data files can be expressed as:

$$C_m^2 = \frac{m(m-1)}{2} \tag{1}$$

Let $s_i(i = 1, 2, \cdots, m)$ be the size of each of the m data files. Then, the total size of the two files in each comparison task can be calculated as:

$$w_{ij} = s_i + s_j(i, j = 1, 2, \cdots, m, i < j) \tag{2}$$

Then, the computing load of the comparison task between file i and file j can be obtained as:

$$c_{ij}(i, j = 1, 2, \cdots, m, i < j) \tag{3}$$

Thus, the total number of multiple comparison tasks between the m data files can be described as:

$$\sum_{i=1}^{m} \sum_{j=i+1}^{m} c_{ij} \tag{4}$$

If the tasks are allocated equally to n nodes, then the theoretical mean number of tasks allocated to each computing node can be expressed as:

$$\frac{\sum_{i=1}^{m} \sum_{j=i+1}^{m} c_{ij}}{n} \tag{5}$$

Then, $x_{kt}(k = 1, 2, \cdots, \frac{m(m-1)}{2}, t = 1, 2, \cdots, n)$ was introduced to specify whether data file k is allocated to node n:

$$x_{kt} = 0 \text{ or } 1, k = 1, 2, \cdots, \frac{m(m-1)}{2}, t = 1, 2, \cdots, n \tag{6}$$

Since a comparison task can only be allocated to one node, we have:

$$\sum_{t=1}^{n} x_{kt} = 1, k = 1, 2, \cdots, \frac{m(m-1)}{2} \tag{7}$$

Let $u_j (j = 1, 2, \cdots, n)$ be the storage capacity of node n and $W_{kt} = w_{ijt}$ be the total size of the two files (file i and file j) in task k that are distributed to node t. Then, the total size of the files distributed to each node in the distributed system should not exceed the storage capacity of that node:

$$\sum_{k=1}^{\frac{m(m-1)}{2}} W_{kt} x_{kt} \leq u_t, t = 1, 2, \cdots, n \tag{8}$$

Let $C_{kt} = c_{ijt}$ be the computing load of the comparison task k between file i and file j that are distributed to node t. Then, the computing load distributed to each node in the distributed system can be expressed as:

$$\sum_{k=1}^{\frac{m(m-1)}{2}} C_{kt} x_{kt} \tag{9}$$

Then, the sum of the absolute difference between the actual and theoretical mean number of tasks actually allocated to each computing node can be expressed as:

$$\sum_{t=1}^{n} \left| \left(\sum_{k=1}^{\frac{m(m-1)}{2}} C_{kt} x_{kt} \right) - \frac{\sum_{i=1}^{m} \sum_{j=i+1}^{m} c_{ij}}{n} \right| \tag{10}$$

Under the constraints of equations (6), (7) and (8), an optimal task allocation model [16] was established to minimize the value of equation (10):

$$\min \sum_{t=1}^{n} \left| \left(\sum_{k=1}^{\frac{m(m-1)}{2}} C_{kt} x_{kt} \right) - \frac{\sum_{i=1}^{m} \sum_{j=i+1}^{m} c_{ij}}{n} \right|$$
$$\text{s.t.} \begin{cases} \sum_{t=1}^{n} x_{kt} = 1, k = 1, 2, \cdots, \frac{m(m-1)}{2} \\ \sum_{k=1}^{\frac{m(m-1)}{2}} W_{kt} x_{kt} \leq u_t, t = 1, 2, \cdots, n \\ x_{kt} = 0 \text{ or } 1, k = 1, 2, \cdots, \frac{m(m-1)}{2}, t = 1, 2, \cdots, n \end{cases} \tag{11}$$

If the objective function in Equation (11) contains nonlinear terms, then established model is a nonlinear programming model. In this case, new decision variables d_t^- and d_t^+ were introduced to Equation (11) to transform the model into the linear form. The variable d_t^- means the actual computing load allocated to a node is greater than the theoretical mean computing load to that node, while the variable d_t^+ has exactly the opposite meaning. The two variables are both numbers greater than or equal to zero. In other words, if the actual computing load allocated to a node is greater than the theoretical mean computing load to that node, the excess load d_t^- should be removed from the node; in the opposite scenario, the insufficient load d_t^+ should be added to the node. In this way, the objective function is changed into finding the minimum sum of $(d_t^- + d_t^+)$ for each node.

After introducing the new decision variables, Equation (11) can be transformed into a linear programming model below.

$$\min \sum_{t=1}^{n} (d_t^- + d_t^+)$$
$$\text{s.t.} \begin{cases} \sum\limits_{t=1}^{n} x_{kt} = 1, k = 1, 2, \cdots, \frac{m(m-1)}{2} \\ \sum\limits_{k=1}^{\frac{m(m-1)}{2}} W_{kt} x_{kt} \le u_t, t = 1, 2, \cdots, n \\ \left(\left(\sum\limits_{k=1}^{\frac{m(m-1)}{2}} C_{kt} x_{kt} \right) - \frac{\sum\limits_{i=1}^{m} \sum\limits_{j=i+1}^{m} c_{ij}}{n} \right) - d_t^- + d_t^+ = 0, t = 1, 2, \cdots, n \\ x_{kt} = 0 \text{ or } 1, k = 1, 2, \cdots, \frac{m(m-1)}{2}, t = 1, 2, \cdots, n \\ d_t^-, d_t^+ \ge 0, t = 1, 2, \cdots, n \end{cases} \tag{12}$$

Since the values of d_t^- and d_t^+ cannot be integers, the model of Equation (12) is a mixed integer linear programming (MILP) model.

3 Design of file allocation algorithm

The MILP model can be solved by commercial solvers like CPLEX [17] and Gurobi [18] and non-commercial approaches like branch and bound method, cutting plane method, branch-cutting plane method and heuristic method [19–21]. Here, the intlinprog function of branch-and-bound steps [16] in the Matlab is selected to solve the MILP in Equation (12). Table 2 lists the comparison tasks and data files in the problem. Among them, branch and bound method is an effective method to solve combinatorial optimization problems. It can get the optimal solution, and the average speed is very fast. Therefore, the idea of branch and bound method is adopted in this paper.

Table 2: Comparison tasks and data files

Task Number	**File1 Number**	**File2 Number**
1	1	2
2	1	3
...	...	...
$m-1$	1	m
M	2	3
$m+1$	2	4
...	...	...
$2*m-3$	2	m
...	...	...
$m(m-1)/2$	$m-1$	m

Then, a file allocation algorithm was designed to cover the following steps [16, 22, 23]:

4 Experimental verification

Four file allocation experiments were carried out in the environment of matlab 2018a [24,25], with the aim to verify our model and algorithm.

Algorithm 1 File allocation algorithm

1 step 1: define and initialize variables
2 define and initialize variables m, n, s, u : $m \leftarrow$ the total number of files, $n \leftarrow$ the total number of nodes, $s \leftarrow [s1, s2, \ldots, sm], u \leftarrow [u1, u2, \leftarrow, un]$;
if s.length==0 or u.length==0 **then**
  $s \leftarrow$ unit matrix whose values are all 1 with m rows and 1 column;
  $u \leftarrow$ unit matrix whose values are all infinite with n rows and 1 column.
end if
3 step 2: calculate the corresponding matrix deci between the tasks and the files
4 define task ordinal variables: $taskno \leftarrow 1$;
5
for all $i = 1$ to m **do**
  for all $j = i + 1$ to m **do**
    $deci(taskno, 1:4) \leftarrow [taskno, i, j, s(i) + s(j)]$;
    $taskno++$;
  end for
end for
6 calculate the theoretical mean number of tasks per node: av_work sum $(deci(,4))/n$;
7 Step 3: set the values for general form of MILP parameters
I) set the value for the objective function variable coefficient matrix f.
Define temporary variables i, j; $i \leftarrow$length$(deci(:,1)) * n; j \leftarrow 2 * n$;
$f \leftarrow$ the matrix with $i + j$ rows and 1 column, the former i row elements are 0, and the latter j row elements are 1.
II) set the value of the variable coefficient matrix$aeq$for the corresponding equality constraint.
for all $i = 1$ to length$(deci(:,1))$ **do**
  for all $j = 1$ to n **do**
    $aeq(i, (i-1) * n + j) \leftarrow 1$;
  end for
end for
for all $j = 1$ to n **do**
  for all $i = 1$ to length$(deci(:,1)$ **do**
    $aeq($length$(deci(:,1)) + j, (i-1) * n + j) \leftarrow deci(i, 4)$;
    $aeq($length$(deci(:,1)) + j, n*$length$(deci(:,1)) + (j-1) * 2 + 1) \leftarrow -1$;
    $aeq($length$(deci(:,1)) + j, n*$length$(deci(:,1)) + j * 2) \leftarrow 1$;
  end for
end for
III) set the value of the resource matrix beq for the corresponding equality constraint
Define temporary variable i; $i \leftarrow$ length$(deci(:,1))$;
$beq \leftarrow (1 + n$ row 1 row matrix, the former i row element is 1, and the latter j row element is av_work;
IV) set the value of the coefficient matrix a for the corresponding inequality constraint
Define temporary variable i; $i \leftarrow$length$(aeq(1,:))$;
$a \leftarrow$ Unit matrix whose values are all 0 with n rows and 1 column.
for all $i = 1$ to n **do**
  for all $j = 1$ to length$(deci(:,1))$ **do**
    $a(i, i + (j-1) * n) \leftarrow deci(j, 4)$;
  end for
end for
V) set the value of the resource matrix b for the corresponding inequality constraint: $b \leftarrow u$;
VI) set the value of the vector $intcon$ for integer variable subscript sequence:
$intcon \leftarrow$1: length$(deci(:,1)) * n$;
VII) set the value of LB for the lower bound and the value of UB for the upper bound. Define the temporary variable $i, j; i \leftarrow$length$(deci(:,1)) * n; j \leftarrow 2 * n$;
$LB \leftarrow$ unit matrix whose values are all 0 with i+j rows and 1 column;
$UB \leftarrow$ the matrix with $i + j$ rows and 1 column, where the former i row elements are 1, and the latter j row elements are + infinity.
8 use the branch and bound intlinprog function to solve the MILP problem, we can get the optimal solution:
$[X, Y] \leftarrow$ intlinprog(f, $intcon$, a, b, aeq, beq, LB, UB);
9 step 4: matrix of comparison task allocation result
Define temporary matrix variables sum, $sum \leftarrow$ unit matrix whose values are all 0 with n rows and 1 column.
for all i=1 to length$(deci(:,1))$ **do**
  for all $j = i + 1$ to n **do**
    if $X((i-1) * n + j) > 0.99999$ **then**
      $sum(j) \leftarrow sum(j) + 1$;
      result$(j, sum(j)) \leftarrow i$;
    end if
  end for
end for

(1) Experiment 1 (Same file size and equal distribution of comparison tasks)

Ten genetic sequence files of the same size (100M) were allocated to five nodes for sequence alignment. As shown in Figure 2, the comparison task distributed to each node has the same computing load and achieves full load balancing. Since $m = 10$ and $n = 5$, we have $m(m-1)\%(2*n) = 0$, that is, each node was distributed with the same number of tasks. The detailed results of experiment 1 are shown in Table 3 below.

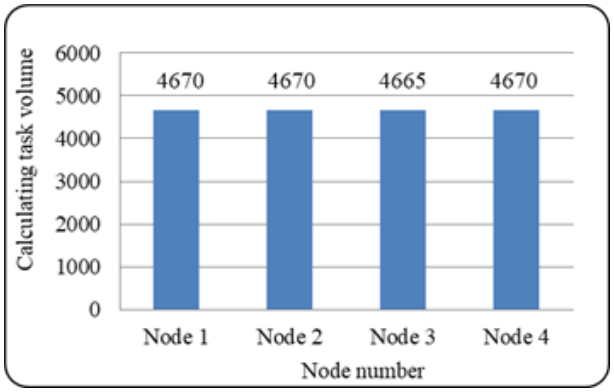


Figure 2: Load distribution to each node in experiment 1

Table 3: Detailed results of experiment 1

Node Number	Task Number	File Number	Task Amount	Calculation Amount
Node1	6, 7, 22, 27, 30, 34, 35, 41, 43	1, 3, 4, 5, 7, 8, 9, 10	9	1800
Node2	3, 9, 13, 16, 17, 25, 28, 31, 36	1, 2, 4, 5, 6, 7, 8, 9, 10	9	1800
Node3	2, 10, 11, 19, 20, 23, 26, 33, 37	1, 2, 3, 4, 5, 6, 8, 9	9	1800
Node4	1, 4, 12, 15, 18, 21, 29, 39, 42	1, 2, 3, 4, 5, 6, 7, 8, 9, 10	9	1800
Node5	5, 8, 14, 24, 32, 38, 40, 44, 45	1, 2, 3, 5, 6, 7, 8, 9, 10	9	1800

(2) Experiment 2 (Same file size and unequal distribution of comparison tasks)

Ten genetic sequence files of the same size (100M) were allocated to four nodes for sequence alignment. Since $m = 10$ and $n = 4$, we have $(m-1)\%(2*n)! = 0$, that is, the nodes cannot achieve complete load balance. In this case, the load distribution to each node is shown in Figure 3. The experimental results (Table 4) show that three of the four nodes were distributed with 2200 tasks while the remaining one was distributed with 2400 tasks. The load between the nodes is basically balanced.

(3) Experiment 3 (Different file sizes and equal distribution of comparison tasks)

Ten genetic sequence files of different sizes (150M; 220M; 180M; 300M; 190M; 95M; 200M; 160M; 320M; 260M) were allocated to five nodes for sequence alignment. Since $m = 10$ and $n = 5$, we have $m(m-1)\%(2*n) = 0$. However, the nodes may not be able to achieve complete load balance due to the difference in file size. In this case, the load distribution to each node is shown in Figure 4. As shown in Table 5, four of the five nodes were distributed with 3735 tasks while the other two with 3775 tasks. The load between the nodes is basically balanced.

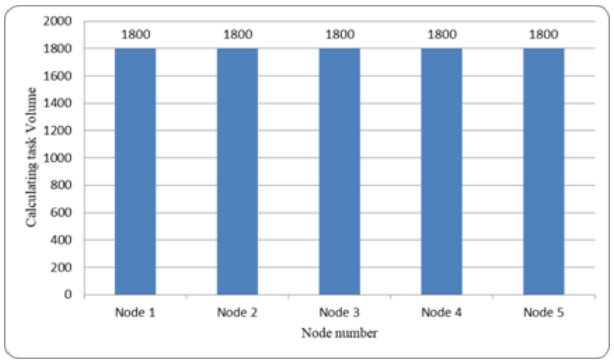


Figure 3: Detailed results of Experiment 2

Table 4: Detailed results of Experiment 2

Node Number	Task Number	File Number	Task Amount	Calculation Amount
Node1	3, 6, 9, 14, 17, 20, 23, 24, 29, 39, 45	1, 2, 3, 4, 6, 7, 9, 10	11	2200
Node2	7, 11, 13, 16, 18, 19, 21, 25, 27, 35, 37, 42	1, 2, 3, 4, 5, 6, 7, 8, 9, 10	12	2400
Node3	4, 5, 8, 22, 31, 33, 34, 38, 41, 43, 44	1, 3, 4, 5, 6, 7, 8, 9, 10	11	2200
Node4	1, 2, 10, 12, 15, 26, 28, 30, 32, 36, 40	1, 2, 3, 4, 5, 6, 7, 8, 10	11	2200

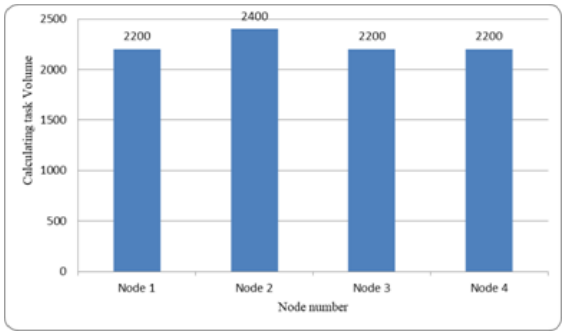


Figure 4: Load distribution to each node in experiment 3

(4) Experiment 4 (Different file sizes and unequal distribution of comparison tasks)

Ten genetic sequence files of different sizes (150M; 220M; 180M; 300M; 190M; 95M; 200M; 160M; 320M; 260M) were allocated to four nodes for sequence alignment. In this case, the load distribution to each node is shown in Figure 5. As shown in Table 6, three of the four nodes were distributed with 4670 tasks while the remaining one with 4665 tasks. The load between the nodes is basically balanced.

The results of the four experiments reveal that our algorithm can always reach load balance between the nodes, whether the files are of the same size, and ensure that the total size of the files distributed to a node never exceeds the storage capacity of that node. Even if load balancing is

Table 5: Detailed results of Experiment 3

Node Number	Task Number	File Number	Task Amount	Calculation Amount
Node1	2, 22, 26, 28, 32, 40, 41, 42, 43	1, 3, 4, 5, 6, 7, 8, 9, 10	9	3735
Node2	5, 10, 12, 13, 18, 20, 31, 35, 36, 45	1, 2, 3, 4, 5, 6, 7, 9, 10	10	3735
Node3	3, 4, 7, 16, 19, 21, 23, 25, 39	1, 2, 3, 4, 5, 6, 7, 8, 9, 10	9	3775
Node4	1, 17, 24, 27, 29, 30, 34, 37	1, 2, 3, 4, 5, 6, 7, 8, 9, 10	8	3735
Node5	6, 8, 9, 11, 14, 15, 33, 38, 44	1, 2, 4, 5, 6, 7, 8, 9, 10	9	3735

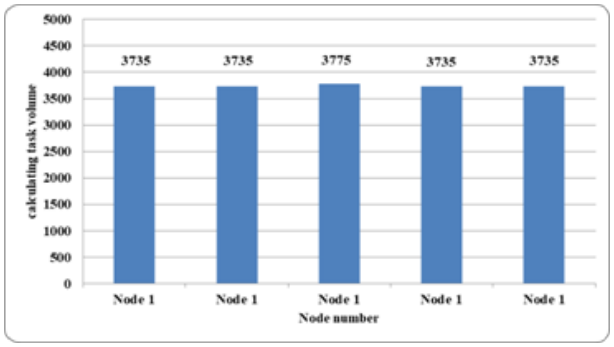


Figure 5: Load distribution to each node in experiment 4

Table 6: Detailed results of Experiment 4

Node Number	Task Number	File Number	Task Amount	Calculation Amount
Node1	1, 8, 11, 17, 19, 29, 32, 33, 41, 45	1, 2, 3, 4, 5, 7, 8, 9, 10	10	4670
Node2	4, 13, 16, 20, 24, 26, 30, 31, 35, 37, 39, 42	1, 2, 3, 4, 5, 6, 7, 8, 9, 10	12	4670
Node3	3, 5, 10, 15, 18, 22, 27, 28, 34, 43, 44	1, 2, 3, 4, 5, 6, 7, 8, 9, 10	11	4665
Node4	2, 6, 7, 9, 12, 14, 21, 23, 25, 36, 38, 40	1, 2, 3, 4, 5, 6, 7, 8, 9, 10	12	4670

theoretically impossible, our algorithm can minimize the load difference between the nodes and approximate the load balance.

Compared with Hadoop-based data allocation strategy, this algorithm can ensure that the comparison tasks have 100% data locality, achieve load balancing between nodes, and improve storage saving and overall computing performance.

5 Conclusions

This paper probes deep into the all-to-all comparison between data files in a large dataset under the environment of a distributed system. After reviewing the problems existing in the existing methods, the author gives a formal mathematical description of the whole comparison problem. In order to achieve load balancing of each node in distributed system, a data file allocation model based on MILP is constructed by using the technology and method of mathematical modeling. Meanwhile, a file allocation algorithm was set up on the Matlab using the intlinprog function of branch-and-bound method. Finally, our model and algorithm were verified through several experiments. The results show that the proposed file allocation strategy can achieve the basic load balance of each node in the distributed system without exceeding the storage capacity of any node, and completely localize the data file. The research findings help to fully utilize the efficiency, stability and scalability of the distributed system to enhance the computing performance of all-to-all comparison.

Acknowledgements. Funding

The work is funded in part by the National Natural Science Foundation of China (NSFC) under Grant No. 61462070, the Doctoral research fund project of Inner Mongolia Agricultural University under Grant No. BJ09-44 and the Inner Mongolia Autonomous Region Key Laboratory of big data research and application for agriculture and animal husbandry.

Author contributions. Conflict of interest

The authors contributed equally to this work. The authors declare no conflict of interest.

Bibliography

[1] Borodin, V.; Bourtembourg, J.; Hnaien, F., Labadie, N. (2018). COTS software integration for simulation optimization coupling: case of ARENA and CPLEX products, *International Journal of Modelling and Simulation*, (5), 1–12, 2018.

[2] Dai, Y.; Wu, W.; Zhou, H.B.; Zhang, J.; Ma, F.Y. (2018). Numerical Simulation and Oprimization of Oil Jet Lubrication for Rotorcraft Meshing Gears, *International Journal of Simulation Modelling*, 17(2), 318–326, 2018.

[3] Dai, Y.; Zhu, X.; Zhou, H.; Mao, Z.; Wu, W. (2018). Trajectory Tracking Control for Seafloor Tracked Vehicle By Adaptive Neural-Fuzzy Inference System Algorithm, *International Journal of Computers Communications & Control*, 13(4), 465–476, 2018.

[4] Deng, J. (2014). Research and Improvement of Mixed Integer Linear Programming Model for Unit Combination, *Nanning: Guangxi University*, 12–16, 2014.

[5] Gao, Y.J. (2017). Research on Data Allocation Strategy for All-to-all Comparison of Large Data Sets, *Taiyuan: Taiyuan University of Technology*, 5–10, 2017.

[6] Guo, J.W.; Li, Y.; Du, L.P.; Zhao, G.F.; Jiang, J.Y. (2014). Research on distributed data mining system based on hadoop platform, *Advances in Intelligent Systems and Computing*, 255, 629–636, 2014.

[7] He, H.; Du, Z.H.; Zhang, W.Z.; Chen, A. (2016). Optimization strategy of Hadoop small file storage for big data in healthcare, *Journal of Supercomputing*, 72(10), 3696–3707, 2016.

[8] Hess, M.; Sczyrba, A.; Egan, R.; Kim, T.W.; Chokhawala, H.; Schroth, G.; Luo, S.; Clark, D.S.; Chen, F.; Zhang, T.; Mackie, R.I.; Pennacchio, L.A.; Tringe, S.G.; Visel, A.; Woyke, T.; Wang, Z.; Rubin, E.M. (2011). Metagenomic discovery of biomass-degrading genes and genomes from cow rumen, *Science*, 331(6016), 463–467, 2011.

[9] Hu, S.R. (1991). Modern supercomputer system, *Journal of computer science*, (1), 47–56, 1991.

[10] Jiao, X.P.; Mu, J.J. (2013). Improved check node decomposition for linear programming decoding, *IEEE Communications Letters*, 17(2), 377–380, 2013.

[11] Liao, J.; Trahay, F.; Xiao, G.; Li, L.; Ishikawa, Y. (2017). Performing initiative data prefetching in distributed file systems for cloud computing, *IEEE Transactions on Cloud Computing*, 5(3), 550–562, 2017.

[12] Mu, R.; Wu, J.J.; Li, N. (2018). MATLAB and mathematical modeling, *Beijing: Science Press*, 63–78, 2018.

[13] MĂźller, E.R.; Carlson, R.C.; Junior, W.K. (2016). Intersection control for automated vehicles with MILP, *IFAC-PapersOnLine*, 49(3), 37–42, 2016.

[14] Nayahi, J.J.V.; Kavitha, V. (2017). Privacy and utility preserving data clustering for data anonymization and distribution on Hadoop, *Future Generation Computer Systems*, 74, 393–408, 2017.

[15] Pitty, S.S.; Karimi, I.A. (2008). Novel MILP models for scheduling permutation flowshops, *Chemical Product and Process Modeling*, 3(1), 35–42, 2008.

[16] Sun, J.Y. (2016). Simulation experiment of operation research model based on MATLAB, *Journal of Shenyang University (Natural Science Edition)*, 28(4), 337–339, 2016.

[17] Schulman, J.; Duan, Y.; Ho, J.; Lee, A.; Awwal, I.; Bradlow, H. (2014). Motion planning with sequential convex optimization and convex collision checking, *International Journal of Robotics Research*, 33(9), 1251–1270, 2014.

[18] Schmidt, B.; Hartmann, C. (2018). Wavepacket: a matlab package for numerical quantum dynamics. ii: open quantum systems, optimal control, and model reduction, *Computer Physics Communications*, 228, 229–244, 2018.

[19] Ubarhande, V.; Popescu, A.; González-Vélez, H. (2015). Novel Data-Distribution Technique for Hadoop in Heterogeneous Cloud Environments, *2015 Ninth International Conference on Complex, Intelligent, and Software Intensive Systems*, 217–224, 2015.

[20] Wang, L.Z.; Tao, J.; Ranjan, R.; Marten, H.; Streit, A.; Chen, J.Y.; Chen, D. (2013). G-Hadoop: MapReduce across distributed data centers for data-intensive computing, *Future Generation Computer Systems*, 29(3), 739–750, 2013.

[21] Yang, X.P.; Zhou, X.G.; Cao, B.Y. (2015). Multi-level linear programming subject to addition-min fuzzy relation inequalities with application in Peer-to-Peer file sharing system, *Journal of Intelligent and Fuzzy Systems*, 28(6), 2679–2689, 2015.

[22] Zhang, Y.F.; Tian, Y.C.; Fidge, C.; Kelly, W. (2016); Data-aware task scheduling for all-to-all comparison problems in heterogeneous distributed systems, *Journal of Parallel & Distributed Computing*, 93(C), 87–101, 2016.

[23] Zhang, Y.F.; Tian, Y.C.; Kelly, W.; Fidge, C. (2017). Scalable and efficient data distribution for distributed computing of all-to-all comparison problems, *Future Generation Computer Systems*, 67, 152–162, 2017.

[24] Zhang, Y.F.; Tian, Y.C.; Kelly, W.; Fidge, C. (2014). A distributed computing framework for All-to-All comparison problems, *IECON 2014 - 40th Annual Conference of the IEEE Industrial Electronics Society*, 2499–2505, 2014.

[25] Zhou, J.X.; Shao, X.M.; Qiao, J.Y.; Zhang, Y.W. (2012). MATLAB from the introduction to proficiency (2nd edition), *Beijing: People's Post and Telecommunications Publishing House*, 35–92, 2012.

INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL
ISSN 1841-9836, e-ISSN 1841-9844, 14(2), 212-219, April 2019.

Performance Analysis of RAW Impact on IEEE 802.11ah Standard Affected by Doppler Effect

A.A. Marwan, D. Perdana, D.D. Sanjoyo

Abdul Aziz Marwan*
Department of Electrical Engineering, Telkom University
Jl. Telekomunikasi No. 1 Terusan Buah Batu 40257, Bandung, Indonesia
*Corresponding author: abdulazizmarwan@student.telkomuniversity.ac.id

Doan Perdana
Department of Electrical Engineering, Telkom University
Jl. Telekomunikasi No. 1 Terusan Buah Batu 40257, Bandung, Indonesia
doanperdana@telkomuniversity.ac.id

Danu Dwi Sanjoyo
Department of Electrical Engineering, Telkom University
Jl. Telekomunikasi No. 1 Terusan Buah Batu 40257, Bandung, Indonesia
danudwj@telkomuniversity.ac.id

Abstract: Internet of Things (IOT) offers a new dimension of technology and information where connectivity is available anywhere, anytime, and for any purpose. IEEE 802.11 Wireless Local Area Network group is a standard that developed to answer the needs of wireless communication technology (WI-Fi). Recently, IEEE 802.11 working group released the 802.11ah technology or Wi-Fi HaLow as a Wi-fi standard. This standard works on the 1 GHz frequency band with a broader coverage area, massive device and the energy efficiency issues. This research addresses, the influence of Doppler Effect using Random Waypoint mobility model on 802.11ah with different RAW slot and RAW slot duration are analyzed. The design of the simulation system is done by changing RAW slot and RAW slot duration. Based on the result, it can be concluded that the overall performance of the network with all of the parameter scenarios is decreasing along with the increasing RAW slot, RAW slot duration, and fluctuation. In the RAW slot = 5 scenario with v = 10 km/h has the worst performance with an average delay which is about 0.128225 s, and average throughput is about 0.284337 Mbps while for RAW slot = 1 with an average PDR which is about 99.1076 %. While in the RAW slot duration = 0.020 s scenario with v = 10 km/h has the worst performance with an average delay which is about 0.135581 s, average throughput is about 0.286828 Mbps, and average PDR which is about 99.3165 %.
Keywords: Restricted Access Window (RAW), IEEE 802.11ah, Random Waypoint, Modulation and Coding Scheme (MCS), Network Simulator 3.

1 Introduction

Nowadays, Internet of Thing (IoT) offers a new dimension in the world of technology and information where connectivity is available wherever, whenever, and for anything. The current global trend of Internet of Thing is very rapidly evolving from the needs of users that want the efficiency of devices in various aspects in order to facilitate the user's own activities [2]. The number of connected devices being the main point of problems in IoT technology itself related to energy efficiency or energy consumption.

The IEEE 802.11 Wireless Local Area Network standard working group operating at 2.4 GHz and 5 Ghz band frequencies is a standard that developed to address the needs for wireless

(Wi-Fi) communication technology problems that have a high data rate, easy to develop and lower value in cost aspect, such as Wireless Sensor Network (WSN) and Machine to Machine (M2M) communication that used in application of military, commercial, health care, monitoring of traffic, and also controlling the inventory [1]. In its development, the IEEE 802.11 working group released 802.11ah or Wi-fi HaLow technology as the new Wi-fi standard. This standard works on a 1 GHz band frequency with broader area coverage, more effective in cost value with an energy efficiency improvement [9]. 802.11ah provides a shortest MAC header, segmented traffic indication map (TIM), restricted access window (RAW), and target wake time (TWT) that support the efficiency and quantity of energy used by stations (STAs) [7].

In its application, 802.11ah technology can accommodate devices or stations in large numbers and every station has their movement pattern such as static or mobile user characteristics. The movement of stations or mobility can affect the performance of the 802.11ah itself. The most commonly used mobility model according to the literature is the Random Waypoint (RMW) model [8]. Firstly, each station will go to the random destination with random speed, move towards the destination, and pause on several times, then moving again towards the coordinates of the destination. Other similar mobility models such as Random Direction model, the Random Walk model, Manhattan and the Gauss-Markov mobility model are also often used in experimental simulations to obtain data that represent real condition network in the world [5]. The term fading means the signal information is being lost on the process of transferring. This can be happened because of some factors such as rapid fluctuation of amplitudes, phases, multipath delays, or user speed. This user speed factor is the Doppler Effect that influencing fading of the signal. The Doppler Effect is a relative motion between the base station and the user which results in random frequency modulation due to different Doppler shifts on each of the multipath components [3].

In this research we discuss about the Doppler Effect with the changing of RAW schemes on IEEE 802.11ah standard network performances using Random Waypoint Mobility model with velocity = 10 km/h. This scenario aims to analyze the performance of RAW impact on 802.11ah and to find the RAW with the worst performance. The RAW schemes are using the RAW slot = 1, 2, 3, 4, 5 and RAW slot duration = 0.005 s, 0.010 s, 0.015 s, 0.020 s. Furthermore, the performance of network is measured using simulation result from Network Simulator 3. The measured output are delay, throughput, and PDR.

2 General description of the used mechanisms

The simulations on this research were performed on Network Simulator 3 release 3.21 with 802.11ah module which has been modified according to [10]. The RAW scenario aims to analyze the Doppler Effect on 802.11ah with different RAW duration and RAW slot. Simulations were performed on 100 nodes with 50 RAW stations. On each number of station, the simulation were performed in two different RAW slot scenarios, and on each RAW slot duration, the simulation were performed in five different RAW slot number as explained in table 1.

The amount of bandwidth and data rate used in the simulation is adjusted to be about twice than the other. Which is MCS 3 (2 MHz bandwidth and 2600 Kbps data rate). Effective and efficient network conditions are required by wireless networks with IEEE 802.11ah standards that capable of allocating stations with large numbers and wide coverage.

In the simulation topology, was placed one Access Point and 100 nodes of STA around it that illustrated in Fig 1. This research focuses on RAW mechanism in MAC layer of 802.11ah standard. The other features such as TIM segmentation and TwT were not implemented.

The simulation has used the Traffic Generator as a sender as well as a receiver packets that will be delivered. Generation of traffic is done by UDP transport protocol because when data is

Figure 1: Topology of Simulation

Table 1: Scenario Explanation

	Slot duration = 0.005 s	Slot = 1
		Slot = 2
		Slot = 3
		Slot = 4
		Slot = 5
	Slot duration = 0.010 s	Slot = 1
		Slot = 2
		Slot = 3
		Slot = 4
		Slot = 5
RAW Scenario	Slot duration = 0.015 s	Slot = 1
		Slot = 2
		Slot = 3
		Slot = 4
		Slot = 5
	Slot duration = 0.020 s	Slot = 1
		Slot = 2
		Slot = 3
		Slot = 4
		Slot = 5

transmitted, data transmission time is more important than its integrity [4], it is in accordance with the needs of delivery data on IoT communications where communication in real time is necessary.

The flowchart system of this research is presented in figure 2. According to the system, after designing the simulation of 802.11ah standard in NS3 environment, traffic generator is

implemented on the simulation. The RAW changing scenario of simulation is designed to collect the data. If the scenarios are succeeded, delay, throughput and PDR data can be collected to be analyzed. Thus, the Doppler Effect influence on network performance can be analyzed for the conclusion.

The output from the simulation in this research is QoS parameters which are as follows [6]:

- Average End to End Delay, which is the average time of delivering the data package from the sender to the receiver.

$$Delay = \frac{\sum Receivedpacketdestination - Packetsentsource}{\sum Packetreceived} \tag{1}$$

- Throughput, which is defined as the speed (rate) effective for transferring the data. Throughput is total number of packets received in bits divided by the number of delivery time.

$$Throughput = \frac{\sum Receivedpacketsize}{\sum Deliverytime} \tag{2}$$

- Packet Delivery Ratio (PDR), which is the ratio between the numbers of packets successfully received and the number of packets sent.

$$PDR = \frac{\sum Totalpacketreceived}{\sum Totalpacketsent} \times 100\% \tag{3}$$

3 Experimental results

The parameters and its description of the simulation are presented in table 2. The output from the simulation is QoS parameters such as delay, throughput, and PDR for RAW slot and RAW slot duration scenario in IEEE 802.11ah standard using Random Waypoint mobility with $v = 10km/h$ which are shown in figure 3 - figure 5.

The Doppler Effect is calculated to find which of the small-scale fading are affected by the RAW slot and RAW slot duration. The impacts are Delay Spread that causing signal power to be weakened and Inter-Symbol Interference (ISI), Doppler Spread that causing signal power to be weakened, and Doppler Shift that causing frequency signal wave to be changed or distorted.

The first influence of Doppler Effect is Delay Spread: Frequency Selective Fading which is about $40 \times 10^{-6}s << 2 \times 10^{-3}s$ or $Ts << \sigma$, where symbol duration is lower than maximum excess delay and $2MHz >> 0,1MHz$ for MCS 3 or $Bs >> Bc$, where bandwidth of signal is higher than channel bandwidth. And the second influence of Doppler Effect is Doppler Spread: Slow Fading which for $v = 10km/h$ is about $21 \times 10^3 \mu s >> 40\mu s$ or $Ts >> Tc_{10}$, where symbol duration is lower than time coherence channel. And the third influence of Doppler Effect is Doppler Shift which for $v = 10km/h$ is about $-8,521Hz$ to $8,521Hz$, where the sender frequency signal wave is distorted.

Figure 3 shows the influence of Doppler Effect in increasing the number of RAW slot and RAW slot duration to the delay that obtained from simulations with MCS 3 (2 MHz bandwidth and 2600 Kbps data rate) using v = 10 km/h user speed. There are some fluctuations in both RAW slot and RAW slot duration. This is the impact of Doppler Spread: Slow Fading causing the delay value in both RAW slot and RAW slot duration to be fluctuated. From the graph above, the highest value of delay that obtained from MCS 3 in RAW slot = 5 with an average delay which is about 0.128225 s while in RAW slot duration = 0.020 s with an average delay which

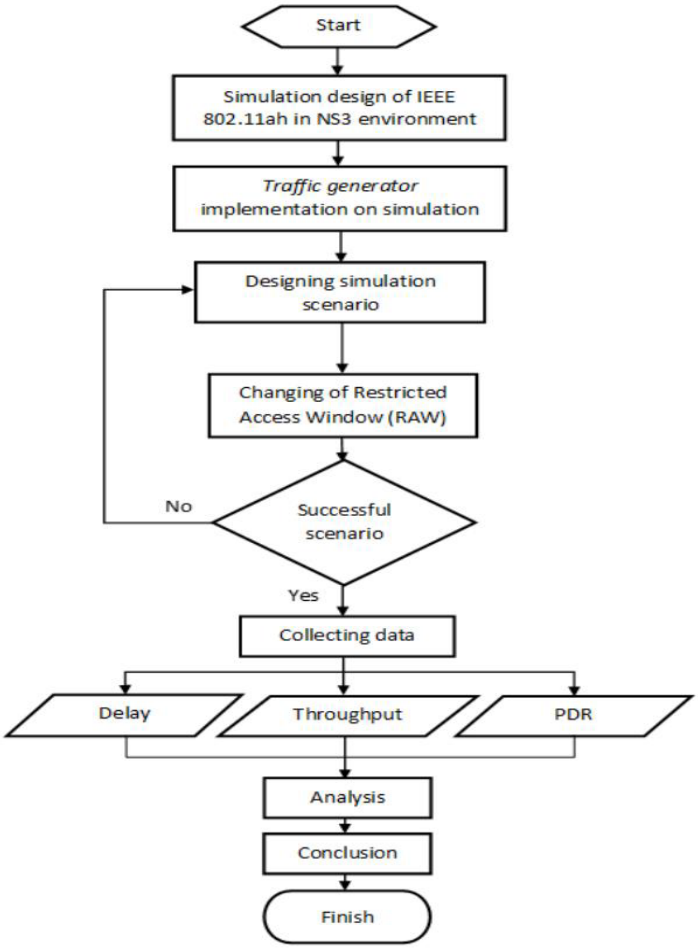


Figure 2: Flowchart System

Table 2: Simulation Parameters

Parameters	**Value**
Physical Layer	WLAN/ IEEE 802.11ah
Transport Layer	UDP
Payload Size	100 Bytes
Rho	100 m
Number of STA	100
Number of RAW STA	50
Number of AP	1
MCS	MCS 3 (2 MHz bandwidth and 2600 Kbps data rate)
RAW Group	1
RAW Slot	1, 2, 3, 4, 5
RAW Slot duration	0.005 s, 0.010 s, 0.015 s, 0.020 s
Mobility Model	Random Waypoint Mobility
User Speed	10 km/h

is about 0.135581 s. Also, the highest fluctuation value of delay that obtained in RAW slot = 5 with an average delay which is about 0.501875 s while in RAW slot duration = 0.020 s with an average delay which is about 0.300501 s. In this scheme, from the result in the terms of average delay, that the network performance is getting lower with the increasing number of RAW slot and RAW slot duration. Meanwhile the fluctuation is getting higher with the increasing number of RAW slot and RAW slot duration. Thus, the higher the fluctuation in delay, the lower the network performance will be in higher RAW slot and RAW slot duration.

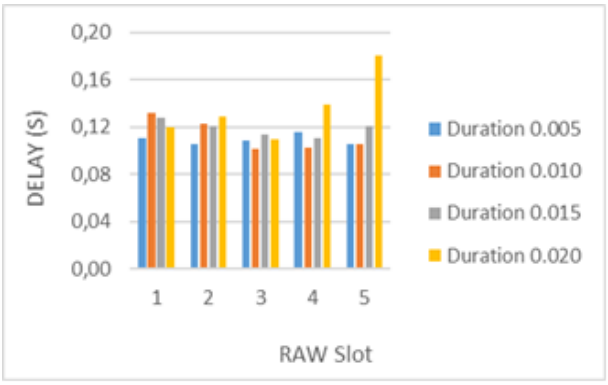


Figure 3: Delay on v = 10 km/h user speed

Figure 4 shows the influence of Doppler Effect in increasing the number of RAW slot and RAW slot duration to the throughput that obtained from simulations with MCS 3 (2 MHz bandwidth and 2600 Kbps data rate) using v = 10 km/h user speed. There are some fluctuations in both RAW slot and RAW slot duration. This is the impact of Doppler Spread: Slow Fading causing the throughput value in both RAW slot and RAW slot duration to be fluctuated. From the graph above, the lowest value of throughput that obtained from MCS 3 in RAW slot = 5 with an average throughput which is about 0.284337 Mbps while in RAW slot duration = 0.020 s with an average throughput which is about 0.286828 Mbps. Also, the highest fluctuation value of throughput that obtained in RAW slot = 5 with an average throughput which is about 0.124824 Mbps while in RAW slot duration = 0.020 s with an average throughput which is about 0.125177 Mbps. In this scheme, from the result in the terms of average throughput, that the network performance is getting lower with the increasing number of RAW slot and RAW slot duration. Meanwhile the fluctuation is getting higher with the increasing number of RAW slot and RAW slot duration. Thus, the higher the fluctuation in throughput, the lower the network performance will be in higher RAW slot and RAW slot duration.

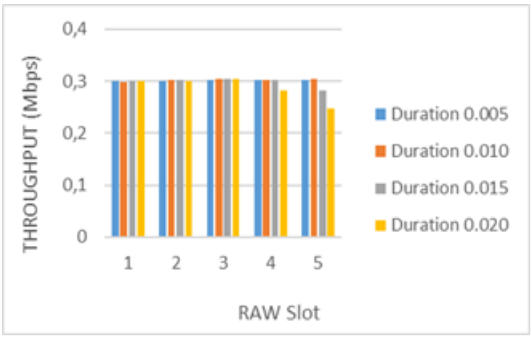


Figure 4: Throughput on v = 10 km/h user speed

Figure 5 shows the influence of Doppler Effect in increasing the number of RAW slot and

RAW slot duration to the PDR that obtained from simulations with MCS 3 (2 MHz bandwidth and 2600 Kbps data rate) using v = 10 km/h user speed. There are some fluctuations in both RAW slot and RAW slot duration. This is the impact of Doppler Spread: Slow Fading causing the PDR value in both RAW slot and RAW slot duration to be fluctuated. From the graph above, the lowest value of PDR that obtained from MCS 3 in RAW slot = 1 with an average PDR which is about 99.1076 % while in RAW slot duration = 0.020 s with an average PDR which is about 99.3165 %. Also, the lowest fluctuation value of PDR that obtained in RAW slot = 1 with an average PDR which is about 0.0006 % while in RAW slot duration = 0.020 s with an average PDR which is about 0.0019 %. In this scheme, from the result in the terms of average PDR, that the network performance is getting lower with the decreasing number of RAW slot and increasing RAW slot duration. Meanwhile the fluctuation is getting lower with the decreasing number of RAW slot and increasing RAW slot duration. Thus, the lower the fluctuation in PDR, the lower the network performance will be in lower RAW slot and in higher RAW slot duration.

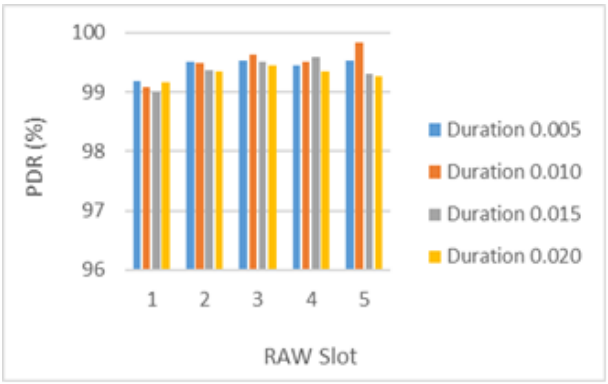


Figure 5: PDR on v = 10 km/h user speed

4 Conclusions

In the RAW scenario, the network performance value will decrease along with the increasing RAW slot and RAW slot duration. This is because the higher the RAW slot and RAW slot duration, the stronger the influence of Doppler Effect which caused the fluctuation. Based on the calculation, the lower the time coherence channel, the stronger the Doppler Spread: Slow Fading.

For delay, the network performance is getting lower with the increasing number of RAW slot and RAW slot duration. Meanwhile the fluctuation is getting higher with the increasing number of RAW slot and RAW slot duration. Thus, the higher the fluctuation in delay, the lower the network performance will be in higher RAW slot and RAW slot duration.

For throughput, the network performance is getting lower with the increasing number of RAW slot and RAW slot duration. Meanwhile the fluctuation is getting higher with the increasing number of RAW slot and RAW slot duration. Thus, the higher the fluctuation in throughput, the lower the network performance will be in higher RAW slot and RAW slot duration.

For PDR, the network performance is getting lower with the decreasing number of RAW slot and increasing RAW slot duration. Meanwhile the fluctuation is getting higher with the decreasing number of RAW slot and increasing RAW slot duration. Thus, the lower the fluctuation in PDR, the lower the network performance will be in lower RAW slot and in higher RAW slot duration.

It can be concluded that for network performance in delay and throughput, the network performance is getting lower with the increasing number of RAW slot and RAW slot duration. While in PDR, the network performance is getting lower with the decreasing number of RAW slot and increasing RAW slot duration. And for fluctuation in delay and throughput, the fluctuation is getting higher with the increasing number of RAW slot and RAW slot duration. While in PDR, the fluctuation is getting higher with the decreasing number of RAW slot and increasing RAW slot duration. Therefore, the fluctuation can be an indicator to analyze which RAW slot and RAW slot duration with the worst network performance where the fluctuation is the impact of the Doppler Effect.

Author contributions. Conflict of interest

The authors contributed equally to this work. The authors declare no conflict of interest.

Bibliography

[1] Aljarrah, E. (2017). Deployment of multi-fuzzy model based routing in RPL to support efficient IoT, *Int. J. Commun. Networks Inf. Secur.*, 9(3), 457-465, 2017.

[2] Khan, I. (2017). Performance Analysis of 5G Cooperative-NOMA for IoT-Intermittent Communication, *Int. J. Commun. Networks Inf. Secur.*, 9(3), 314-322, 2017.

[3] Mitra, A. (2009). *Lecture Notes on Mobile Communication: A Curriculum Development Cell Project Under QIP*, IIT Guwahati, India, 2009.

[4] Olariu, C. (2013). *Quality of Service Support for Voice over IP in Wireless Access Networks*, Waterford Institute of Technology, 2013.

[5] Perdana, D; Munadi, R.; Manurung, R.C. (2017). Performance Evaluation of Gauss-Markov Mobility Model in Hybrid LTE-VANET Networks, *Telkomnika(Telecommunication Computing Electronics and Control)*, 15(2), 606-621, 2017.

[6] Putra, M.A.P.; Perdana, D.; Negara, R.M. (2017). Performance Analysis of Data Traffic Offload Scheme on Long Term Evolution (LTE) and IEEE 802.11AH, *Telkomnika(Telecommunication Computing Electronics and Control)*, 15(4), 1659-1665, 2017.

[7] Raeesi, O.; Pirskanen, J.; Hazmi, A.; Levanen, T.; Valkama, M. (2014). Performance evaluation of IEEE 802.11ah and its restricted access window mechanism, *Proc. IEEE ICC Workshops*, 460-466, 2014.

[8] Rupinder, K.; Gurpreet, S. (2014). Survey of Various Mobility Models in VANETs, *Int. J. Eng. Comput. Sci.*, 3(3), 4073-4080, 2014.

[9] Sun, W.; Choi, M.; Choi, S. (2014). IEEE 802.11ah: A Long Range 802.11 WLAN at Sub 1 GHz, *J. ICT Stand.*, 2(2), 83-108, 2014.

[10] Tian, L.; Deronne, S.; Latre, S.; Famaey, J. (2016). Implementation and Validation of an IEEE 802.11ah Module for ns-3, *Conf. Work. ns3 (WNS3), Seattle, USA*, 49-56, 2016.

INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL
ISSN 1841-9836, e-ISSN 1841-9844, 14(2), 220-232, April 2019.

Extended TODIM Method for MADM Problem under Trapezoidal Intuitionistic Fuzzy Environment

H.P. Ren, M.F. Liu, H. Zhou

Haiping Ren*
Teaching Department of Basic Subjects
Jiangxi University of Science and Technology,330013 Nanchang, China
*Corresponding author: 9520060004@jxust.edu.cn

Manfeng Liu
The Collaborative Innovation Center
Jiangxi University of Finance and Economics,330013 Nanchang, China
liumanfeng@sina.com

Hui Zhou
School of Mathematics and Computer Science
Yichun University,336000 Yichun, China
huihui7978@126.com

Abstract: In actual decision making process, the final decision result is often affected by decision maker's psychological behavior, however, for the multiple attribute decision making (MADM) problem in which attributes values are expressed with trapezoidal intuitionistic fuzzy numbers, there are few literatures considering the decision maker's behavior factors in decision making process. For this case, this paper first proposes a new distance measure of TIFNs and a new ranking method which considers decision maker's attitude behavior, and then develops an extended TODIM decision making method. Finally an example is given to illustrate the validity and practicability of the proposed method.
Keywords: TODIM method, trapezoidal intuitionistic fuzzy number, multiple attribute decision making, ranking method.

1 Introduction

In recent years, with the increasing complexity of the managerial decision making environment, many managerial decision-making problems contain qualitative properties which are difficult to quantify. Zadeh's fuzzy sets have been greatly successful in dealing with fuzzy management decision making problems [3, 4, 18, 20, 22, 35]. Zadeh's fuzzy set is characterized by a single scale (membership), which can only characterize the support and opposition of the two aspects of the evidence. But some decision making problems have ambiguous hesitant phenomenon with respect to evaluation of information, and Zadeh's fuzzy set is hard or difficult to depict these situations. Therefore, many scholars developed Zadeh's fuzzy set, and intuitionistic fuzzy (IF) set is one of the most famous fuzzy sets among them. Originally proposed by Atanassov in 1986 [1], IF sets can well describe the hesitation and uncertainty of judgment through the addition of a non-membership parameters, which can describe the vague characters of things comprehensively. Then IF sets have become a powerful and effective tool in dealing with uncertain or vague information in actual applications. In dealing with ambiguity and uncertainty, IF sets are more flexible and practical than fuzzy sets, and thus they have been applied widely in decision making.

Because of the complexity and uncertainty of objective things and the limitation of decision maker's knowledge, membership and non-membership functions are sometimes difficult to represent by using the precise numbers. But interval number can be very useful to describe this kind

of case, so Atanassov and Gargov [2] extended IF sets to interval-valued IF sets. Some scholars put forward the concept of continuous IF numbers to describe an uncertain quantity or a difficult quantification number on the basis of the concept of IF set. Grzegrorzewski [10] extended IF sets to the continuous case of IF numbers . Nehi and Maleki [21] put forward trapezoidal intuitionistic fuzzy number (TIFN), and defined the corresponding operation rules, which caused great concern in the academic community. Shu [27] proposed the definition of triangular intuitionistic fuzzy number, which is a special example of TIFN, and they put it to the application of fault tree analysis, base on these research, Wang and Zhang [31] further expanded it and gave the definition of a generalized TIFN. Different from the definitions of IF sets, TIFN is added to a trapezoidal fuzzy number, which makes the membership degree and the non-membership degree no longer be only a fuzzy concept *Good* or *Excellent*; then, the assessment information given by the decision makers can be expressed exactly. Comparing with IF sets, they have more attractive explanation, and are easy to be quantified and executed by the decision maker, and thus they have more theoretical value in the field of decision science [8, 14, 16, 29].

At present, the theory and application of fuzzy numbers, such as triangular intuitionistic fuzzy number, trapezoidal intuitionistic fuzzy number, have been received great attention. But most of the existing decision-making methods do not consider the influence of the behavior of the decision-makers in the decision process, because they are assumed that the decision maker is completely rational. However, the actual decision-making process is often accompanied by the different psychological behavior of the decision-makers and the attitude of the risk and other factors of behavior. Kahneman and Tversky [13] proposed the prospect theory, which can describe the decision maker's psychological behavior. Based on prospect theory, Gomes and Lima ([6, 7]) developed a new MADM method named TODIM method, which has made many successful applications, such as material evaluation [34], green supplier selection [26], logistics outsourcing [30] etc. Fan et al. [5] proposed an extension of TODIM (H-TODIM) to solve the hybrid MADM problems in which attribute values have three forms: crisp numbers, interval numbers and fuzzy numbers. Qin [23] proposed a generalization of the TODIM method under triangular intuitionistic fuzzy environment. Ren et al. [25] extended the TODIM method to deal with the MADM problem in which attribute values are expressed with Pythagorean fuzzy numbers. Zhang et al. [33] developed the TODIM method to solve the MADM problem in which the attribute values are expressed with neutrosophic numbers. In this paper, we will develop a new extension of TODIM method to solve the MADM problem in which attribute values are expressed with TIFNs, and an application example is used to illustrate the validity and practicability of the proposed method.

2 Preliminary knowledge

2.1 Definitions of TIFNs

Firstly, we recall the definition of the TIFN and the related theory. In order to use the concept of IF sets to define an uncertain number or difficult to quantify the amount, Grzegrorzewski [10] extended the IF sets to the continuous case of the IF sets, and gave the following definition:

Definition 8. Let R be the set of real numbers, A is called an IF number in R , if its membership function $\mu_A(x)$ and non-membership function $\nu_A(x)$ are respectively defined as follows ([10]) :

$$\mu_A(x) = \begin{cases} 0, & x < a_1 \\ f_A(x), & a_1 \le x \le a_2 \\ 1, & a_2 \le x \le a_3 \\ g_A(x), & a_3 \le x \le a_4 \\ 0, & a_4 < x \end{cases}$$

and

$$\nu_A(x) = \begin{cases} 1, & x < b_1 \\ h_A(x), & b_1 \le x \le b_2 \\ 0, & b_2 \le x \le b_3 \\ k_A(x), & b_3 \le x \le b_4 \\ 1, & a_b < x \end{cases}$$

where $0 \le \mu_A(x) + \nu_A(x) \le b_2, a_i, b_i \in R, i = 1, 2, 3, 4$, and they satisfy $b_1 \le a_1 \le b_2 \le a_2 \le b_3 \le a_3 \le b_4 \le a_4$. The four functions $f_A(x), g_A(x), h_A(x)$ and $k_A(x)$ are real value aunctions defined in interval [0,1]. The functions $f_A(x)$, $k_A(x)$ are non-decreasing coctinuous functions and $g_A(x)$, $h_A(x)$ are non-increasing continuous functions.

On the basis of Grzegrorzewski's IF numbers, Nehi [21] developed the TIFN in 2005, which is given in Definition 2.

Definition 9. Let $b_1 \le a_1 \le b_2 \le a_2 \le b_3 \le a_3 \le b_4 \le a_4 \in R$. A fuzzy number A is called a TIFN, if its membership function $\mu_A(x)$ and non-membership function $\nu_A(x)$ are respectively defined as follows([21]):

$$\mu_A(x) = \begin{cases} 0, & x < a_1 \\ \frac{x-a_1}{a_2-a_1}, & a_1 \le x \le a_2 \\ 1, & a_2 \le x \le a_3 \\ \frac{a_4-x}{a_4-a_3}, & a_3 \le x \le a_4 \\ 0, & a_4 < x \end{cases}$$

and

$$\nu_A(x) = \begin{cases} 1, & x < b_1 \\ \frac{x-b_1}{b_2-b_1}, & b_1 \le x \le b_2 \\ 0, & b_2 \le x \le b_3 \\ \frac{b_4-x}{b_4-b_3}, & b_3 \le x \le b_4 \\ 1, & a_b < x \end{cases}$$

We denote A by $A =< (a_1, a_2, a_3, a_4), (b_1, b_2, b_3, b_4) >$.

Definition 10. For two TIFNs $A_1 =< (a_{11}, a_{12}, a_{13}, a_{14}), (b_{11}, b_{12}, b_{13}, b_{14}) >$ and $A_2 =< (a_{21}, a_{22}, a_{23}, a_{24}), (b_{21}, b_{22}, b_{23}, b_{24}) >$, the operational laws are defined as follows [2]:

(1) $A_1 + A_2 =< (a_{11}+a_{21}, a_{12}+a_{22}, a_{13}+a_{23}, a_{14}+a_{24}), (b_{11}+b_{21}, b_{12}+b_{22}, b_{13}+b_{23}, b_{14}+b_{24}) >$

(2) $kA_1 =< (ka_{11}, ka_{12}, ka_{13}, ka_{14}), (kb_{11}, kb_{12}, kb_{13}, kb_{14}) >$, for $k > 0$

Usually, $\alpha-$cut set is a very effective tool for describing the number of fuzzy numbers [11], and TIFNs have two classes $\alpha-$cut sets: $(A^+)_\alpha$ and $(A^-)_\alpha$.

Definition 11. Let $A =< (a_1, a_2, a_3, a_4), (b_1, b_2, b_3, b_4) >$ be a TIFN, then the two classes $\alpha-$cut sets $(A^+)_\alpha$ and $(A^-)_\alpha$ are defined as follows, respectively:

$$(A^+)_\alpha = \{x \in R | \mu_A(x) \ge \alpha\}$$

$$(A^-)_\alpha = \{x \in R | 1 - \nu_A(x) \ge \alpha\}$$

According to Definition 4, each $\alpha-$cut set is a closed interval,thus they can be denoted as $(A^+)_\alpha = [(A_L^+)_\alpha, (A_U^+)_\alpha]$ and $(A^-)_\alpha = [(A_L^-)_\alpha, (A_U^-)_\alpha]$, where

$$(A_L^+)_\alpha = inf\{x \in R|\mu_A(x) \geq \alpha\},$$
$$(A_U^+)_\alpha = sup\{x \in R|\mu_A(x) \geq \alpha\},$$
$$(A_L^-)_\alpha = inf\{x \in R|1-\nu_A(x) \geq \alpha\},$$
$$(A_U^-)_\alpha = sup\{x \in R|1-\nu_A(x) \geq \alpha\}.$$

Then, for a TIFN $A =< (a_1, a_2, a_3, a_4), (b_1, b_2, b_3, b_4) >$, we can easily derive the following results:

$$(A^+)_\alpha = [(A_L^+)_\alpha, (A_U^+)_\alpha] = [a_1 + (a_2 - a_1)\alpha, a_4 - (a_4 - a_3)\alpha],$$
$$(A^-)_\alpha = [(A_L^-)_\alpha, (A_U^-)_\alpha] = [b_1 + (b_2 - b_1)(1-\alpha), b_4 - (b_4 - b_3)(1-\alpha)],$$

2.2 A novel distance measure of TIFNs

In the follow we will develop a novel distance of TIFNs based on the following distance measure between two fuzzy numbers proposed by Grzegorzewski in 1998.

Lemma 2.1. *For two any fuzzy numbers A and B, and the corresponding $\alpha-$cut sets are respectively $[(A_L^+)_\alpha, (A_U^+)_\alpha]$ and $[(A_L^-)_\alpha, (A_U^-)_\alpha]$, then the distance measure between them is defined as [9].*

$$d(A,B) = (\int_0^1 ((A_L)_\alpha - (B_L)_\alpha)^2 d\alpha + \int_0^1 ((A_U)_\alpha - (B_U)_\alpha)^2 d\alpha)^{1/2}.$$

Inspired by the Lemma 1, we define the following distance measure for two arbitrary TIFNs and as follows:

$$d(A,B) = \tfrac{1}{2}(\int_0^1 ((A_L^+)_\alpha - (B_L^+)_\alpha)^2 + ((A_U^+)_\alpha - (B_U^+)_\alpha)^2 d\alpha)^{1/2}$$
$$+\tfrac{1}{2}(\int_0^1 ((A_L^-)_\alpha - (B_L^-)_\alpha)^2 + ((A_U^-)_\alpha - (B_U^-)_\alpha)^2 d\alpha)^{1/2}$$

It is easy to prove that the new distance measure can satisfy the non negativity, symmetry and triangle inequality. By straightforward calculation, we can get Theorem 1.

Theorem 1. *Let $A_1 =< (a_{11}, a_{12}, a_{13}, a_{14}), (b_{11}, b_{12}, b_{13}, b_{14}) >$ and $A_2 =< (a_{21}, a_{22}, a_{23}, a_{24}), (b_{21}, b_{22}, b_{23}, b_{24}) >$ be two TIFNs,then the distance measure between A_1 and A_2 is defined as follows:*

$$d(A_1, A_2) = \frac{1}{2}\left[(I_1 + I_2)^{1/2} + (I_3 + I_4)^{1/2}\right] \tag{1}$$

Proof: Here

$$\begin{aligned} I_1 &= \int_0^1 ((A_{1L}^+)_\alpha - (A_{2L}^+)_\alpha)^2 d\alpha \\ &= \int_0^1 [(a_{21} - a_{11}) + (a_{22} - a_{21} - a_{12} + a_{11})\alpha]^2 d\alpha \\ &= \int_0^1 [x + (y-x)\alpha]^2 d\alpha \\ &= x^2 + x(y-x) + \tfrac{1}{3}(y-x)^2 \\ &= \tfrac{1}{3}(x^2 + xy + y^2) \\ &= \tfrac{1}{3}((a_{21} - a_{11})^2 + (a_{21} - a_{11})(a_{22} - a_{12}) + (a_{22} - a_{12})^2) \end{aligned}$$

where $x = a_{21} - a_{11}, y = a_{22} - a_{12}$. Similarly, we have

$$\begin{aligned} I_2 &= \int_0^1 ((A_{1U}^+)_\alpha - (A_{2U}^+)_\alpha)^2 d\alpha \\ &= \tfrac{1}{3}((a_{23} - a_{13})^2 + (a_{23} - a_{13})(a_{24} - a_{14}) + (a_{24} - a_{14})^2) \end{aligned}$$

$$I_3 = \int_0^1 \left((A_{1L}^-)_\alpha - (A_{2L}^-)_\alpha\right)^2 d\alpha$$
$$= \frac{1}{3}((b_{21}-b_{11})^2 + (b_{21}-b_{11})(b_{22}-b_{12}) + (b_{22}-b_{12})^2)$$
$$I_4 = \int_0^1 \left((A_{1U}^-)_\alpha - (A_{2U}^-)_\alpha\right)^2 d\alpha$$
$$= \frac{1}{3}((b_{23}-b_{13})^2 + (b_{23}-b_{13})(b_{24}-b_{14}) + (b_{24}-b_{14})^2)$$

Then by lemma 1, we can derive the conclusion (1). □

3 A new ranking function of TIFNs

In actual decision making process, the final decision result is often affected by the different attitudes of decision-makers, although many scholars have already considered the influence of different attitude index for the MADM problems in which attributes values are expressed with interval number and triangular fuzzy number [17, 24, 28]. However, for the MADM problem in which attributes values are expressed with TIFNs, and there is no literature considering the decision maker's attitude in decision making process. Thus, we take the decision maker's mentality into the decision-making process, and put forward a new TIFN ranking method.

Definition 12. Let $\tilde{a} = [a^L, a^U]$ be an interval fuzzy number, $M_{\tilde{a}} = 0.5(a^L + a^U)$ and $D_{\tilde{a}} = 0.5(a^U - a^L)$. $F_{\tilde{a}}(\lambda) : [0,1] \to \tilde{a}$ is a function of parameter with the following form:

$$F_{\tilde{a}}(\lambda) = M_{\tilde{a}} + (2\lambda - 1)D_{\tilde{a}} = (1-\lambda)a^L + \lambda a^U.$$

Here, the parameter λ is called attitude index of interval number $\tilde{a}$.

Remark 3.1. Apparently, $F_{\tilde{a}}(\lambda)$ is a monotonic increasing function on interval$[0,1]$. When an attribute is a benefit type attribute, i.e. the value of it is the-larger-the-better. When $\lambda = 0$,then $F_{\tilde{a}}(0) = a^L = [a^L, a^L]$ is smaller than the fuzzy number $\tilde{a} = [a^L, a^U]$,thus for benefit type attribute, the parameter $\lambda = 0$ demonstrates a pessimistic attitude. Similarly, $\lambda = 1$ demonstrates an optimistic attitude and $\lambda = 0.5$ demonstrates a moderate attitude.

Lemma 3.1. *Let $\tilde{a} = (a^L, a^M, a^U)$ be a triangular fuzzy number, and for any real number $\alpha \in [0,1]$, $\alpha - cut$ set of $\tilde{a}$ can be easily derived as follows [15]:*

$$\tilde{a}_\alpha = [a^L(\alpha), a^U(\alpha)] = [a^L + (a^M - a^L)\alpha, a^U - (a^U - a^M)\alpha]$$

Remark 3.2. For two arbitrary triangular fuzzy number $\tilde{a}$ and $\tilde{b}$, $\alpha - cut$ sets are often used to compare them. Considering the $\alpha - cut$ sets of triangular fuzzy numbers are still interval numbers, and α is an arbitrary value in interval $[0,1]$,to eliminate the arbitrariness and reflect the decision maker's attitude behavior, Ren and Liu [24]developed a new ranking function of triangular fuzzy number considering with attitude of decision maker(s) motivated by Definition 5.

Definition 13. Let $\tilde{a} = (a^L, a^M, a^U)$ be a triangular fuzzy number, then for any parameter $\lambda \in [0,1]$, the function $F(\tilde{a}, \lambda)$ is a new ranking function of triangular fuzzy number considering attitude of decision maker(s) with the following formula [24]:

$$F(\tilde{a}, \lambda) = \int_0^1 (1-\lambda)a^L(\alpha) + \lambda a^U(\alpha) d\alpha$$

Obviously, $F(\tilde{a}, \lambda)$ can be rewritten as the following form:

$$F(\tilde{a}, \lambda) = \int_0^1 (1-\lambda)(a^L + (a^M - a^L)\alpha) + \lambda(a^U - (a^U - a^M)\alpha) d\alpha$$
$$= [(1-\lambda)a^L + a^M + \lambda a^U]/2$$

Let $r \in [0,1]$, then according to Definition 6, Ren and Liu [28] gave the following rule for comparing two triangular fuzzy numbers $\tilde{a} = (a^L, a^M, a^U)$ and $\tilde{b} = (b^L, b^M, b^U)$:
(i) For arbitrary $\lambda \in [0,1]$, if $F(\tilde{a}, \lambda) \leq F(\tilde{b}, \lambda)$,then $\tilde{a}$ is smaller than $\tilde{b}$, and noted $\tilde{a} \leq \tilde{b}$;
(ii)For arbitrary $\lambda \in [0,1]$, if $F(\tilde{a}, \lambda) = F(\tilde{b}, \lambda)$,then $\tilde{a}$ is equal to $\tilde{b}$, and noted $\tilde{a} = \tilde{b}$;
(iii)For arbitrary $\lambda \in [0, r]$,if $F(\tilde{a}, \lambda) \leq F(\tilde{b}, \lambda)$,while when $\lambda \in [r, 1]$, $F(\tilde{a}, \lambda) \geq F(\tilde{b}, \lambda)$; then for the decision maker whose attitude is pessimistic, the ranking result is $\tilde{a} \leq \tilde{b}$, which for the decision maker whose attitude is optimistic, the ranking result is $\tilde{a} \geq \tilde{b}$.

Motivated by Definition 6, we will develop a new ranking function of trapezoidal intuitionistic fuzzy number defined in Definition 7.

Definition 14. Let $A =< (a_1, a_2, a_3, a_4), (b_1, b_2, b_3, b_4) >$ be a TIFN, and $p(\alpha)$ is a real function defined on $[0,1]$, then a new ranking function $F(A, \lambda)$ including the attitude behavior of decision maker is defined as follows:

$$F(A,\lambda) = \tfrac{1}{2}\int_0^1 (1-\lambda)(A_L^+)_\alpha + \lambda(A_U^+)_\alpha dP(\alpha) \\ +\tfrac{1}{2}\int_0^1 (1-\lambda)(A_L^-)_\alpha + \lambda(A_U^-)_\alpha dP(\alpha)$$

Particularly, if $P(\alpha) = \alpha^{r+1}$, then we can get

$$F(A,\lambda) = \tfrac{1}{2(r+2)}[(1-\lambda)(a_1+b_1) + (r+1)(1-\lambda)(a_2+b_2) \\ +(r+1)\lambda(a_3+b_3) + \lambda(a_4+b_4)]$$

Remark 3.3. If $r = 0, \lambda = 1/2$, then $F(A, \lambda)$ is as same as that ranking function of Ye [32]. Similar discussion with Remark 1, the parameter λ is the attitude index. Then the ranking function $F(A, \lambda)$ can reflect the attitude behavior, and thus it can better depict the actual decision process with the help of different values of λ than that ranking function of Ye [32].

Definition 15. For two given TIFNs A_1 and A_2 , and $r \in [0,1]$, the relationship of A_1 and A_2 can be defined as follows:
(i) For any $\lambda \in [0,1]$, if $F(A_1, \lambda) \leq F(A_2, \lambda)$,then A_1 is smaller than A_2, and noted $A_1 \leq A_2$;
(ii)For any $\lambda \in [0,1]$, if $F(A_1, \lambda) = F(A_2, \lambda)$,then A_1 is equal to A_2, and noted $A_1 = A_2$;
(iii)For any $\lambda \in [0, r]$,if $F(A_1, \lambda) \leq F(A_2, \lambda)$,while when $\lambda \in [r, 1]$, $F(A_1, \lambda) \geq F(A_2, \lambda)$; then for the decision maker whose attitude is pessimistic, the ranking result is $A_1 \leq A_2$, while for the decision maker whose attitude is optimistic, the ranking result is $A_1 \geq A_2$.

4 Extended TODIM method for MADM under TIFN environment

For a given MADM problem, let $X = \{x_1, x_2, \cdots, x_m\}$ be a possible alternatives set, and $O = \{o_1, o_2, \cdots, o_n\}$ be the evaluation attribute set.$D = \{D_1, D_2, \cdots, D_s\}$ is the expert set. Suppose the rating of x_i $(i = 1, 2, \cdots, m)$with respect to $o_j (j = 1, 2, \cdots, n)$given by expert D_k $(k = 1, 2, \cdots, s)$is a linguistic term noted by $\tilde{s}_{ij}^k$, which belongs to the linguistic terms set { Absolutely low, Low, Fairly low, Fairly high, High, Absolutely high }. Then the MADM problem can be expressed with matrices $\tilde{S}^k = (s_{ij}^k)_{m\times n}, k = 1, 2, ..., s$.
Let $w = (w_1, w_2, ..., w_n)^T$ be the attribute weight vector, and each element w_j represents the degree of importance of attribute, which can be given by decision maker or determined by some weighting methods, such as AHP method or entropy weighting method.

The calculation steps of the extended TODIM method considering the decision maker's attitude are given as follows:

Step 1. According to Table 1 [32],$\tilde{s}_{ij}^k$ can be transformed with TIFNs $\tilde{a}_{ij}(k)$,$i = 1, 2, ..., m, j = 1, 2, ..., n$ and $k = 1, 2, ..., s$.

Table 1: Linguistic terms and corresponding TIFNs

Linguistic terms	**TIFNs**
Absolutely low (AL)	$< (0.001, 0.001, 0.001, 0.001), (0.001, 0.001, 0.001, 0.001) >$
Low (L)	$< (0.0, 0.1, 0.2, 0.3), (0.0, 0.1, 0.2, 0.3) >$
Fairly low (FL)	$< (0.1, 0.2, 0.3, 0.4), (0.0, 0.2, 0.3, 0.5) >$
Medium (M)	$< (0.3, 0.4, 0.5, 0.6), (0.2, 0.4, 0.5, 0.7) >$
Fairly high (FH)	$< (0.5, 0.6, 0.7, 0.8), (0.4, 0.6, 0.7, 0.9) >$
High (H)	$< (0.7, 0.8, 0.9, 1.0), (0.7, 0.8, 0.9, 1.0) >$
Absolutely high (AH)	$< (1.0, 1.0, 1.0, 1.0), (1.0, 1.0, 1.0, 1.0) >$

The linguistic terms decision matrices $\tilde{S}^k = (s_{ij}^k)_{m\times n}$ are transformed into trapezoidal intuitionistic fuzzy decision matrices $\tilde{A}^k = (\tilde{a}_{ij}^k)_{m\times n}$ $(k = 1, 2, ..., s)$.

Step 2. Let $\tilde{s}_{ij}$ be the total score of alternative x_i with respect to attribute o_j given by all decision makers, and it is defined as

$$\tilde{a}_{ij} = \frac{1}{s}\sum_{k=1}^{s} \tilde{a}_{ij}^k. \tag{2}$$

Step 3. Determine the weights of evaluation attributes. Using the Definition 7 and $P(\alpha) = \alpha^{r+1}$, we can get the intuitionistic fuzzy sorting function matrix $F(\lambda) = (F(\tilde{a}_{ij}, \lambda))_{m\times n}$,where $F(\tilde{a}_{ij}, \lambda)$ is the ranking function of fuzzy number $\tilde{a}_{ij}$ considering with the attitude of decision maker. For the maximum TIFN $\tilde{a}^* =< (1, 1, 1, 1), (1, 1, 1, 1) >$, $F(\tilde{a}^*, \lambda) = 1$.

Now, we will propose a new weighting method by means of the proposed ranking function. The reasonable weight should be the minimum of the total deviation of the alternative $x_i(i = 1, 2, \cdots, m)$ and the positive ideal solution $\tilde{a}^*$. Therefore, we can establish the following optimization model:

$$\begin{aligned} &\min G(w) = \sum_{j=1}^{n}\sum_{i=1}^{m} w_j(1 - F(\tilde{a}_{ij})) \\ &s.t. \quad \begin{cases} w \in H \\ \sum\limits_{j=1}^{n} w_j = 1 \\ w_j \geq 0,\ j = 1, 2, ..., n \end{cases} \end{aligned} \tag{3}$$

By solving the Eq. (3), the optimal solution $w^* = \arg\max S$ is chosen as the optimal attribute weights.

Step 4. Calculate TODIM score as follows:
(i) Calculate $w_{rc} = \frac{w_c}{w_r}$,where the value w_{rc} represents the weight value of criteria r divided by the weight of the reference point c,and $w_r = \max\limits_{1\leq c\leq n}\{w_c\}$.

(ii) For given value of attitude index λ ,calculate

$$\phi_c(x_i, x_j) = \begin{cases} \sqrt{\frac{d(x_i,x_j)}{w_{rc}}}, & F(x_{ic}, \lambda) - F(x_{jc}, \lambda) > 0 \\ 0 & F(x_{ic}, \lambda) - F(x_{jc}, \lambda) = 0 \\ -\frac{1}{\theta}\sqrt{\frac{d(x_i,x_j)}{w_{rc}}}, & F(x_{ic}, \lambda) - F(x_{jc}, \lambda) < 0 \end{cases}$$

Here $d(x_i, x_j) = \sum_{c=1}^{n} w_{rc} d(\tilde{a}_{ic}, \tilde{a}_{jc})$. Here the parameter θ is an important parameter in prospect theory, and $\theta > 1$ shows that the individual is losses aversion, and $\theta < 1$ shows the individuals are attenuated when facing the losses [19]. Here we set $\theta = 2.25$, which is the most often used value of θ in prospect theory.

(iii) Let $\delta(A_i, A_j) = \sum_{c=1}^{n} \phi_c(A_i, A_j)$, $i, j = 1, 2, ..., m$, calculate the comprehensive evaluation index value:

$$\xi_i = \frac{\sum_{j=1}^{m} \delta(A_i, A_j) - \min\limits_{1\le i\le m} \sum_{j=1}^{m} \delta(A_i, A_j)}{\max\limits_{1\le i\le m} \sum_{j=1}^{m} \delta(A_i, A_j) - \min\limits_{1\le i\le m} \sum_{j=1}^{m} \delta(A_i, A_j)}, i = 1, 2, \cdots, m.$$

Step 5. Rank the alternatives according to $\xi_i (i = 1, 2, \cdots, m)$ in decreasing order.

5 Applied example

Suppose that a company wants to invest a large amount of money in the best options(Herrera and Herrera-Viedma [12]; [32]). There are four parallel alternatives: x_1 (a car company), x_2 (a food company), x_3 (a computer company), x_4 (an arms company) and three evaluation attributes o_1(the risk analysis), o_2 (the growth analysis), and o_3 (the environmental impact analysis). The risk investment company now employs four experts to evaluate these four alternative enterprises. The evaluation values are expressed with linguistic terms, and the corresponding trapezoidal intuitionistic fuzzy evaluation decision matrices are listed in Table 2 to Table 4. Our task is to choose the best investment plan by the method presented in this paper.

Table 2: Linguistic evaluation values given by expert 1

Alternatives	o_1	o_2	o_3
x_1	M	M	FL
x_2	FH	FH	M
x_3	M	FH	M
x_4	H	M	FL

The specific calculation steps of the proposed decision making method considering with the psychological behavior of the decision makers are given below:

Step 1. The linguistic terms decision matrices $\tilde{S}^k = (s_{ij}^k)_{m\times n}$ are transformed into $\tilde{A}^k = (\tilde{a}_{ij}^k)_{m\times n}$ $(k = 1, 2, ..., s)$ and given in Table 5 to Table 7.

Step 2. The evaluation information of the expert group is gathered and expressed with decision matrix $\tilde{A} = (\tilde{a}_{ij})_{m\times n}$, which is shown in Table 8.

Table 3: Linguistic evaluation values given by expert 2

Alternatives	o_1	o_2	o_3
x_1	FL	M	L
x_2	FH	H	M
x_3	M	FH	FL
x_4	H	FH	FL

Table 4: Linguistic evaluation values given by expert 3

Alternatives	o_1	o_2	o_3
x_1	M	FH	FL
x_2	M	FH	M
x_3	FH	FH	M
x_4	H	H	M

Table 5: Trapezoidal intuitionistic fuzzy decision matrix given by expert 1

Alternatives	o_1	o_2	o_3
x_1	<(0.3,0.4,0.5,0.6), (0.2,0.4,0.5,0.7)>	<(0.3,0.4,0.5,0.6), (0.2,0.4,0.5,0.7)>	<(0.1,0.2,0.3,0.4), (0.0,0.2,0.3,0.5)>
x_2	<(0.5,0.6,0.7,0.8), (0.4,0.6,0.7,0.9)>	<(0.5,0.6,0.7,0.8), (0.4,0.6,0.7,0.9)>	<(0.3,0.4,0.5,0.6), (0.2,0.4,0.5,0.7)>
x_3	<(0.3,0.4,0.5,0.6), (0.2,0.4,0.5,0.7)>	<(0.5,0.6,0.7,0.8), (0.4,0.6,0.7,0.9)>	<(0.3,0.4,0.5,0.6), (0.2,0.4,0.5,0.7)>
x_4	<(0.7,0.8,0.9,1.0), (0.7,0.8,0.9,1.0)>	<(0.3,0.4,0.5,0.6), (0.2,0.4,0.5,0.7)>	<(0.1,0.2,0.3,0.4), (0.0,0.2,0.3,0.5)>

Table 6: Trapezoidal intuitionistic fuzzy decision matrix given by expert 2

Alternatives	o_1	o_2	o_3
x_1	<(0.1,0.2,0.3,0.4), (0.0,0.2,0.3,0.5)>	<(0.3,0.4,0.5,0.6), (0.2,0.4,0.5,0.7)>	<(0.0,0.1,0.2,0.3), (0.0,0.1,0.2,0.3)>
x_2	<(0.5,0.6,0.7,0.8), (0.4,0.6,0.7,0.9)>	<(0.7,0.8,0.9,1.0), (0.7,0.8,0.9,1.0)>	<(0.3,0.4,0.5,0.6), (0.2,0.4,0.5,0.7)>
x_3	<(0.3,0.4,0.5,0.6), (0.2,0.4,0.5,0.7)>	<(0.5,0.6,0.7,0.8), (0.4,0.6,0.7,0.9)>	<(0.1,0.2,0.3,0.4), (0.0,0.2,0.3,0.5)>
x_4	<(0.7,0.8,0.9,1.0), (0.7,0.8,0.9,1.0)>	<(0.5,0.6,0.7,0.8), (0.4,0.6,0.7,0.9)>	<(0.1,0.2,0.3,0.4), (0.0,0.2,0.3,0.5)>

Step 3. In order to facilitate the comparison with the results of Ye [32], here we also assume that the attribute weights are known with $w_1 = 0.3490$, $w_2 = 0.3020$ and $w_3 = 0.3490$.

Step 4. For given attitude index value $\lambda = 1/2$ and $r = 0$, the comprehensive evaluation values of the extended TODIM method are calculated as

$$\xi_1 = 0, \xi_2 = 1.0000, \xi_3 = 0.7421, \xi_4 = 0.4370,$$

Table 7: Trapezoidal intuitionistic fuzzy decision matrix given by expert 3

Alternatives	o_1	o_2	o_3
x_1	<(0.3,0.4,0.5,0.6), (0.2,0.4,0.5,0.7)>	<(0.5,0.6,0.7,0.8), (0.4,0.6,0.7,0.9)>	<(0.1,0.2,0.3,0.4), (0.0,0.2,0.3,0.5)>
x_2	<(0.3,0.4,0.5,0.6), (0.2,0.4,0.5,0.7)>	<(0.5,0.6,0.7,0.8), (0.4,0.6,0.7,0.9)>	<(0.3,0.4,0.5,0.6), (0.2,0.4,0.5,0.7)>
x_3	<(0.5,0.6,0.7,0.8), (0.4,0.6,0.7,0.9)>	<(0.5,0.6,0.7,0.8), (0.4,0.6,0.7,0.9)>	<(0.3,0.4,0.5,0.6), (0.2,0.4,0.5,0.7)>
x_4	<(0.7,0.8,0.9,1.0), (0.7,0.8,0.9,1.0)>	<(0.7,0.8,0.9,1.0), (0.7,0.8,0.9,1.0)>	<(0.3,0.4,0.5,0.6), (0.2,0.4,0.5,0.7)>

Table 8: Evaluation information of the expert group

x_i	o_1	o_2	o_3
x_1	<(0.2333,0.3333,0.4333, 0.5333),(0.1333,0.3333, 0.4333,0.6333) >	<(0.3667,0.4667,0.5667, 0.6667),(0.2667,0.4667, 0.5667,0.7667) >	<(0.0667,0.1667,0.2667, 0.3667),(0.0000,0.1667, 0.2667,0.4333) >
x_2	<(0.4333,0.5333,0.6333, 0.7333),(0.3333,0.5333, 0.6333, 0.8333)>	<(0.5667,0.6667,0.7667, 0.5667),(0.5000,0.6667, 0.7667, 0.9333)>	<(0.3000,0.4000,0.5000, 0.6000),(0.2000,0.4000, 0.5000,0.7000)>
x_3	<(0.3667,0.4667,0.5667, 0.6667),(0.2667,0.4667, 0.5667,0.7667) >	<(0.5000,0.6000,0.7000, 0.8000),(0.4000,0.6000, 0.7000,0.9000)>	<(0.2333,0.3333,0.4333, 0.5333),(0.1333,0.3333, 0.4333,0.6333)>
x_4	<(0.7000,0.8000,0.9000, 1.0000),(0.7000,0.8000, 0.9000,1.0000)>	<(0.5000,0.6000,0.7000, 0.8000),(0.4333,0.6000, 0.7000,0.8667)>	<(0.1667,0.2667,0.3667, 0.4667),(0.0667,0.2667, 0.3667,0.5667)>

Step 5. Based on the values of $\xi_i (i = 1, 2, 3, 4)$, the ranking order of the alternatives is obtained as

$$x_2 > x_3 > x_4 > x_1,$$

and x_2 is the best alternative.This result is in agreement with the one obtained in (Ye [32]).

6 Conclusion

Interactive multiple criteria (TODIM) decision method is developed on the basis of the prospect theory, which can describe the psychological behavior of human under uncertain environment, and has been successfully applied to many MADM problems. TODIM method is easier than prospect theory in processing fuzzy numbers, and some authors have already developed it to solve MADM problems in which the attributes values are expressed with crisp numbers, triangular fuzzy numbers, intuitionistic fuzzy numbers, Pythagorean fuzzy and neutrosophic numbers. However, there is no research on the trapezoidal intuitionistic fuzzy environment, and the main work of this paper is to extend TODIM method to solve MADM problems under TIFN environment. First, the article proposes a new class of distance measure of TIFNs, the distance measure can better measure the difference between two TIFNs. Then a new ranking function of TIFNs is introduced, which can take into account the decision-makers' attitude with an attitude index. Finally, the extended TODIM method is put forward to solve the MADM problem in which the attribute evaluation values are expressed with TIFNs. The advantage of this method lies

in the decision making process which can take into account the decision maker's mentality and the decision maker's perceived value of the gain and loss, so that the decision-making process is more consistent with the objective reality. The proposed distance measure and ranking function can also be used to other MADM methods when the attribute values are expressed with TIFNs.

Funding

The authors would like to thank the support of the National Natural Science Foundation of China (No.71661012) and Science & Technology Research Project of Jiangxi Educational Committee (No. GJJ170496 and No. GJJ180829).

Author contributions. Conflict of interest

The authors contributed equally to this work. The authors declare no conflict of interest.

Bibliography

[1] Atanassov, K.T. (1986). Intuitionistic Fuzzy Sets, *Fuzzy Sets and Systems*, 20, 87-96,1986.

[2] Atanassov, K. T.; Gargov, G. (1989). Interval-valued intuitionistic fuzzy sets, *Fuzzy Sets and Systems*, 31(3), 343–349, 1989.

[3] Bozic, M.; Ducic, N.; Djordjevic, G.; Slavkovic, R. (2017). Optimization of Wheg Robot Running with Simulation of Neuro-Fuzzy Control, *International Journal of Simulation Modelling*, 16(1), 19–30, 2017.

[4] Dalman, H.; Gazel, N.; Sivri M.(2016). A Fuzzy set-based approach to multi-objective multi-item solid transportation problem under uncertainty, *InternationalJournal of Fuzzy Systems*, 18(4), 716–729, 2016.

[5] Fan, Z.P.; Zhang, X.; Chen, F.D., Liu, Y. (2013). Extended TODIM method for hybrid multiple attribute decision making problems, *Knowlege-Based Systems*, 42, 40–48, 2013.

[6] Gomes, L. F. A. M.; Lima, M. M. P. P.(1992). From modeling individual preferences to multicriteria ranking of discrete alternatives: a look at prospect theory and the additive difference model, *Foundations of Computing and Decision Sciences*, 17, 171–184, 1992.

[7] Gomes, L. F. A. M.; Rangel, L. A. D. (2009). An application of the TODIM method to the multicriteria rental evaluation of residential properties, *European Journal of Operational Research*, l193, 204–211, 2009.

[8] Govindan, K.; Jepsen, M. B.(2016). Supplier risk assessment based on trapezoidal intuitionistic fuzzy numbers and ELECTRE TRI-C: a case illustration involving service suppliers, *Journal of the Operational Research Society*, 67(2), 339–376, 2016.

[9] Grzegorzewski, P. (1998). Metrics and orders in space of fuzzy numbers, *Fuzzy sets and Systems*, 97(1), 83–94, 1998.

[10] Grzegrorzewski, P. (2003). In ISTANBULL,The Hamming distance between intuitionistic fuzzy sets, *Springer, Proc. of the IFSA 2003 World Congress*, 35–38, 2003.

[11] Gu,Y. K.;Zhu, F. L.; Tang, S. Y.(2012). Reliability analysis method based on fuzzy probability importance degree, *Journal of Jiangxi University of Science and Technology*, 33(5), 51–55, 2012.

[12] Herrera, F.; Viedma, E. H. (2000). Linguistic decision analysis: steps for solving decision problems under linguistic information, *Fuzzy Sets and Systems*, 115(1), 67–82, 2000.

[13] Kahneman D.; Tversky, A. (1979). Prospect theory: an analysis of decision under risk, *Econometrica*, 47, 263–292, 1979.

[14] Lakshmana, G. N. V.; Jeevaraj, S.; Dhanasekaran, P. (2016.; A linear ordering on the class of trapezoidal intuitionistic fuzzy numbers, *Expert Systems with Applications*, 60, 269–279, 2016.

[15] Li, D.-F. (2003). Fuzzy Multiobjective Many-Person Decision Making and Games. *National Defense Industry Press*, 2003.

[16] Li, X. H.; Chen, X. H. (2015). Multi-criteria group decision making based on trapezoidal intuitionistic fuzzy information, *Applied Soft Computing*, 30, 454–461, 2015.

[17] Liou, T. S.; Wang, M. J. (1992). Fuzzy weighted average:an improved algorithm, *Fuzzy Sets and Systems*, 49(3), 307–315, 1992.

[18] Llopis-Albert, C.; Palacios-Marqus, D.; Merigo, J. M. (2016). Decision making under uncertainty in environmental projects using mathematical simulation modeling, *Environmental Earth Sciences*, 75(19), 1320–1330, 2016.

[19] Lourenzutti, R.; Krohling, R. A. (2014). The Hellinger distance in multicriteria decision making: an illustration to the TOPSIS and TODIM methods, *Expert Systems with Applications*, 41(9), 4414–4421, 2014.

[20] Mousavi, M.; Yap, H. J.; Musa, S. N.; Dawal, S. Z. M. (2017). A Fuzzy Hybrid GA-PSO Algorithm for Multi-Objective AGV Scheduling in FMS, *International Journal of Simulation Modelling*, 16(1), 58–71, 2017.

[21] Nehi, H. M.; Maleki, H. R. (2005). In Athens, Intuitionistic fuzzy numbers and it's applications in fuzzy optimization problem, *Proceedings of the 9th WSEAS International Conference On Systems, 2005*,1–5, 2005.

[22] Pham, V. N.; Long, T. N.; Pedrycz W. (2016). Interval-valued fuzzy set approach to fuzzy co-clustering for data classification, *Knowledge-Based Systems*, 107, 1–13, 2016.

[23] Qin, Q.; Liang, F.; Li, L.; Wu, G. F. (2017). A TODIM-based multi-criteria group decision making with triangular intuitionistic fuzzy numbers, *Applied Soft Computing*, 55, 93–10, 2017.

[24] Ren, H. P.; Liu, M. F. (2014). VIKOR method for MADM problem with triangular fuzzy number considering behavior of decision maker, *ICIC Express Letters, Part B: Applications*, 5(3), 879–884, 2014.

[25] Ren, P.; Xu, Z.; Gou, X.(2016). Pythagorean fuzzy TODIM approach to multi-criteria decision making, *Applied Soft Computing*, 42, 246–259, 2016

[26] Sang, X.; Liu, X. (2016). An interval type-2 fuzzy sets-based TODIM method and its application to green supplier selection, *Journal of the Operational Research Society*, 67(5), 722–734, 2016.

[27] Shu, M. H.; Cheng, C. H.; Chang, J. R.(2006). Using intuitionistic fuzzy sets for fault-tree analysis on printed circuit board assembly, *Microelect ronics Reliability*, 46(12), 2139–2148, 2006.

[28] Wan, S. P. (2009); Method of attitude index for interval multi-attribute decision-making, *Control and Decision*, 24(1), 35–38, 2009.

[29] Wan, S. P.; Dong, J. Y. (2015). Power geometric operators of trapezoidal intuitionistic fuzzy numbers and application to multi-attribute group decision making, *Applied Soft Computing*, 29, 153–168, 2015.

[30] Wang, J.; Wang, J. Q.; Zhang, H. Y. (2016). A likelihood-based TODIM approach based on multi-hesitant fuzzy linguistic information for evaluation in logistics outsourcing, *Computers & Industrial Engineering*, 99, 287–299, 2016.

[31] Wang, J. Q.; Zhang, Z. (2008). Programming method of multi-criteria decision-making based on intuitionistic fuzzy number with incomplete certain information, *Control and Decision*, 23(10), 1145–1148, 2008.

[32] Ye, J. (2012). Multicriteria group decision-making method using vector similarity measures for trapezoidal intuitionistic fuzzy number, *Group Decision Negotiation*, 21, 519–530, 2012.

[33] Zhang, M.; Liu, P.; Shi L. (2016). An extended multiple attribute group decision-making TODIM method based on the neutrosophic numbers, *Journal of Intelligent & Fuzzy Systems,* 30(3), 1773–1781, 2016.

[34] Zindani, D.; Maity, S. R. Bhowmik, S.; Chakraborty, S. (2017). A material selection approach using the TODIM (TOmada de Decisao Interativa Multicriterio) method and its analysis, *International Journal of Materials Research*, 108(5), 345–354, 2017.

[35] Zhu, B.; Xu, Z.S. (2014). A fuzzy linear programming method for group decision making with additive reciprocal fuzzy preference relations, *Fuzzy Sets and Systems*, 246, 19–33, 2014.

INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL
ISSN 1841-9836, e-ISSN 1841-9844, 14(2), 233-252, April 2019.

Routing in WSNs Powered by a Hybrid Energy Storage System through a CEAR Protocol Based on Cost Welfare and Route Score Metric

R. Senthilkumar, G.M. Tamilselvan, S. Kanithan, N. Arun Vignesh

R. Senthilkumar*
Department of Electrical and Electronics Engineering,
Roever Enginering College, Perambalur,
Tamil Nadu- 621212
*Corresponding author: rajsen1985@gmail.com

G.M. Tamilselvan
Electronics and Communication Engineering,
Bannari Amman Institute of Technology, Sathyamangalam,
Tamil Nadu- 638401
tamiltamil@rediffmail.com

S. Kanithan
Department of Electronics and Communication Engineering,
AVS College of Technology-Salem,
Tamilnadu
kani.rex@gmail.com

N. Arun Vignesh
Department of Electronics and Communication Engineering,
GRIET, Hyderabad
arunvignesh44@gmail.com

Abstract: Implementing a low cost, power efficient and high performance routing protocol in wireless sensor networks (WSNs) is an important requirement for transmitting a packet through network. In this paper we propose, a new cost and energy aware routing protocol (CEAR) that works based on the two metrics such as cost welfare metric and route score metric.A hybrid electrical energy storage (HEES) framework which holds numerous banks of heterogeneous electrical energy storage (EES) components to be specific battery and a ultra-capacitor is used for providing energy to the network exhibit in the WSN for routing. The simulation results shows that our proposed routing protocol routes the packet efficiently by choosing the best path that also reduces the cost and routes the packet with reduced power consumption. The quantitative metrics in terms of packet delivery ratio of 0.93, average end to end delay of 110 secs, packet loss ratio of 0.75, average throughput attained of 250 bits/sec and efficiency of 98-99.9% overpowers the performance of our proposed work.
Keywords: Hybrid energy storage system; CEAR protocol, routing; cost welfare metric, route score metric.

1 Introduction

In a wide spectrum of applications such as wireless sensor networks (WSNs), the energy harvesting technology from the solar panels is a primary issue, which is to be tackled for grant effectiveness. The harvested energy should be stored and controlled using appropriate devices to power the sensor nodes present in the wireless networks [4]. Batteries are currently used for powering in most wireless sensors, where the power requirements are modest. The primary

batteries are usually chosen for their high energy densities, low leakage rates and low cost. For sensor node applications, lifetimes of battery is at least a year is desirable [8].A hybrid energy framework, shaped by consolidating various energy storage systems (e.g., battery, ultra-capacitor, flywheel, power module, and so on.) and generators (e.g., smaller scale turbine, wind turbine, PV board, and so on.), has turned out to be a solution to meet the energy/power prerequisites with enhanced adaptability, unwavering quality and cost proficiency [5].Subsequent to the framework arrangement and conduct of the hybrid framework are turning out to be more unpredictable, the streamlined administration and control of a hybrid energy framework is still a testing errand [20]. To accomplish the ideal solution for this energy administration issue, research activity in recent years has focused on the design and development of novel hybrid energy storage systems that use renewable energy technologies [3].

Hybrid energy storage systems (HESS) are essential, when single renewable energy technologies need to be utilized in a more effective way and if they need to be more appealing from a financial point of view [10]. The significant point of interest of this hybrid energy storage systems is that they can, at the same time guarantee distinctive types of energy, for instance electricity, hot water, heating/cooling capacity, and so on, which is by and large an interest in diverse sorts of building offices [25]. Another ecological aspect of this HESS is they are critical and absolutely contemplated with respect to present atmosphere issues and the ones that anticipate us. Regardless of the way that the structure of HESSs is reasonably fundamental, those systems have been generally connected [6].

The fundamental idea in HESS is to utilize ultra-capacitors (UCs) as a right hand energy storage system along with the batteries to enhance the execution of the whole energy storage systems, in terms of effectiveness, unwavering quality, and element reaction [29]. UCs give quick and effective energy conveyance and long cycle-existence without any concoction response included. Likewise, the state of charge (SOC) of a UC can be precisely acquired in light of the fact that its SOC is relative to the square of the cell voltage [7]. The scope of UC working temperature (- 40 to $+70^{O}$C) is too more extensive than that of batteries. The essential disservice of UCs is their moderately low energy thickness contrasted with batteries [9], [17]. Hence hybridization of batteries (and/or power modules) and UCs is considered in nature to be the best utilization of UCs without a doubt applications [2]. Such stored energies are efficiently managed and supplied to the wireless sensor nodes when there is a demand of power [13]- [14].

In order to operate the self-sustainable WSNs in isolated places, the energy aware routing protocol forwards the data packets through network, while extending the lifetime of network [11]. Before its capacity falls below than 80% of its initial rate capacity, the nodes are still limited by the number of charge/discharge cycles of a Rechargeable Battery (RB) [16], [25]. Wireless communications require more energy consumption than sensing and computing tasks [28]. The HESS is composed by RB and a super-capacitor (SC). A power management device is utilized to control all the operations in the energy system [23].

To reduce the power consumption in the sensor nodes and manage the power between the sensor nodes, this paper we propose a Cost and Energy Aware Routing (CEAR) protocol, which reduces the cost and power consumption by the evaluation of two metrics namely cost welfare metric and route source metric. The performance evaluation are conducted based on packet delivery ratio, average end to end delay, packet loss, packet loss ratio, average throughput. The remaining of this paper is organized as follows: Section 2 presents some the recent works related to our proposed work. Section 3 presents the detailed explanation about our proposed methodology. Section 4 presents the results obtained by our proposed routing protocol followed by the conclusion in section 5.

2 Related work

Some of the most recent works related to our proposed routing method is explained as follows:-

Zhang, Bo, et al. [27] described a broadly useful multi-hop WSN architecture capable of supporting time-critical CPS systems utilizing energy harvesting. At that point they exhibited a set of Harvesting Aware Speed Selection (HASS) algorithms. That boosts the base energy save for every one of the nodes in the network, accordingly guaranteeing highly strong performance under emergency or fault driven circumstances. At that point they introduced an ideal centralized solution, alongside a distributed solution and actualized a CPS-specific trial procedure.

Arabinda Nandha et.al [19] explained a work distance, energy and the angle formed by the node with the base station are taken as input parameters. The Mamdani fuzzy inference system is used to select the chance of a sensor node to become a cluster head (CH). A-Star search algorithm is used to locate an optimum route from source to sink node. The data packets are routed on the selected path. The A-Star search algorithm finds the shortest route path from source node to sink node. This approach is admissible, complete, and optimal one as it uses A-Star algorithm.

Selvi et.al [24] described a technique called the rule based clustering for routing model provides better performance in terms of network lifetime than the other existing techniques since they consume more energy during the formation of clusters and finding the shortest path. Moreover, additional overhead on the cluster head selection is tackled also using rules in this proposed model in an efficient manner by building balanced clusters. The main advantage of the proposed approach is that it extends the lifetime of the network and increases the throughput, energy efficiency, link quality, and scalability.

Roshani H.Padyal et.al [21] implemented an Opportunistic based neighbor coverage routing. In this hop-by-hop opportunistic routing is done by selecting a forwarding node on the base of the current load on the node, energy, and neighbor coverage information. In this choosing of the reliable node in the path, an establishment is based on load information table. Proposed protocols minimize energy consumption and enhance network performance as compared to existing routing protocol.

Suneet Kumar Gupta et.al [22] discussed an energy efficient and balanced route generation algorithm is proposed with considering both energy efficacy and energy balancing issues. Here, we consider the distance and residual energy of the nodes as energy efficiency parameters and energy is balanced by diverting the incoming traffic to other nodes having comparably lower incoming traffic. To develop the routing schedule, we have applied Genetic Algorithm which can quickly compute the routing schedule as per the current state of the network. It is observed that the performance of proposed algorithm is better than existing algorithm in terms of first node die and energy consumption in the network.

Nilanjan Mukherjee and Dani Strickland [18] actualized a particular boost multilevel buck converter based topology that coordinate the half breed second life batteries with the grid tie inverter. A reasonable module based distributed control architecture was introduced to totalize each converter modules according to its characteristics autonomously. The converter and control architecture were observed to be adaptable to coordinate different batteries with an inverter dc link. Modeling, investigation and test approval were performed on a solitary stage secluded half breed battery energy storage system model to comprehend the operations of the control technique with different cross breed battery configurations.

Abeywardana, DB Wickramasinghe,et al. [1] built up a battery-supercapacitor based cross breed energy storage system (HESS) comprising of two DC/AC support converters, battery, supercapacitors, matrix connection, state of charge (SOC) estimation and an associated control

systems were tentatively verified and further moved forward. The switching frequency current swell component in both battery and supercapacitor currents were reduced by stage shifted interleaved operation. In the meantime, the control system keeps up the supercapacitor voltage in pre-characterized esteem and the battery's SOC. The estimation can be finished by a broadened Kalman channel kept up within the specified SOC limits.

Qing Xie et al. [26] formally described the worldwide charge allocation issues in HEES systems, in particular, conveying a specified level of incoming energy to a subset of destination EES (Electrical Energy Storage) banks, with the goal that efficiency of most extreme charge allocation was achieved. To set the worldwide charge allocation efficiency, the issue was figured as a mixed integer nonlinear program with the objective function, where the constraints capturing key requirements and features of the system such as the energy conservation law, power conversion losses in the chargers, rate capacity and self-discharge effects in the EES components.

Huang, et al. [12] actualized an energy sharing controller in view of distributed battery energy storage system architecture. The re-designed DC-DC control stage and the energy sharing controller are used to achieve SOC balancing among the battery cells while in the meantime giving DC bus voltage direction to whatever is left of the system or load. As a result, there is no requirement for two autonomous converter systems for cell SOC balancing and DC bus voltage control. This prompts reduced design complexity of the battery energy storage system. The proposed energy sharing controller tends to the battery cells' SOC imbalance issue from the root by changing the discharge/charge rate of each battery cell.

Zhang, Bo, et al. [27] described a broadly useful multi-hop WSN architecture capable of supporting time-critical CPS systems utilizing energy harvesting. At that point they exhibited aset of Harvesting Aware Speed Selection (HASS) algorithms. That boosts the base energy save for every one of the nodes in the network, accordingly guaranteeing highly strong performance under emergency or fault driven circumstances. At that point they introduced an ideal centralized solution, alongside a distributed solution and actualized a CPS-specific trial procedure.

3 Routing in WSN using CEAR protocol

Mobilizing the natural resources such as solar energy in the maximum range and storing the energy which is used to power the sensor nodes present in the wireless sensor network (WSN) is done by the design of hybrid accumulator that consist of a battery and a ultra-capacitor is implemented in the previous work. Also the stored energy between battery and the ultra-capacitor is controlled and managed using an adaptive power organizing algorithm in the previous work. The power supplied to the sensor nodes by means of the battery and ultra-capacitor in the hybrid accumulator depending on the demand of power. The success of wireless sensor networks and their pervasive use is somehow constrained by energy supply which generally provided by batteries. The network lifetime depends on the balance between energy consumption and reliable delivery of data packets. The nodes in the wireless networks powered by a hybrid energy storage system (HESS) require a specific management unit to control the different energy flows. However, limited battery life has been a barrier for widespread deployment of wireless networks. To overcome such issues, in this paper we proposed a Cost and Energy Aware Routing (CEAR) protocol for efficient and inexpensive routing. The architecture of our proposed method is shown in figure 1. From figure 1 we can see that the sensor nodes present in the wireless sensor network is powered by means of renewable energy source such as solar energy extracted from the PV panels. The extracted energy is stored and supplied to the WSN through the hybrid energy storage systems which is a combination of a battery and a capacitor. The supplied energy is consumed by the nodes to perform routing and using CEAR protocol the best path which reduces the power consumption and the cost is selected and the data packets get transmitted through

such paths efficiently.

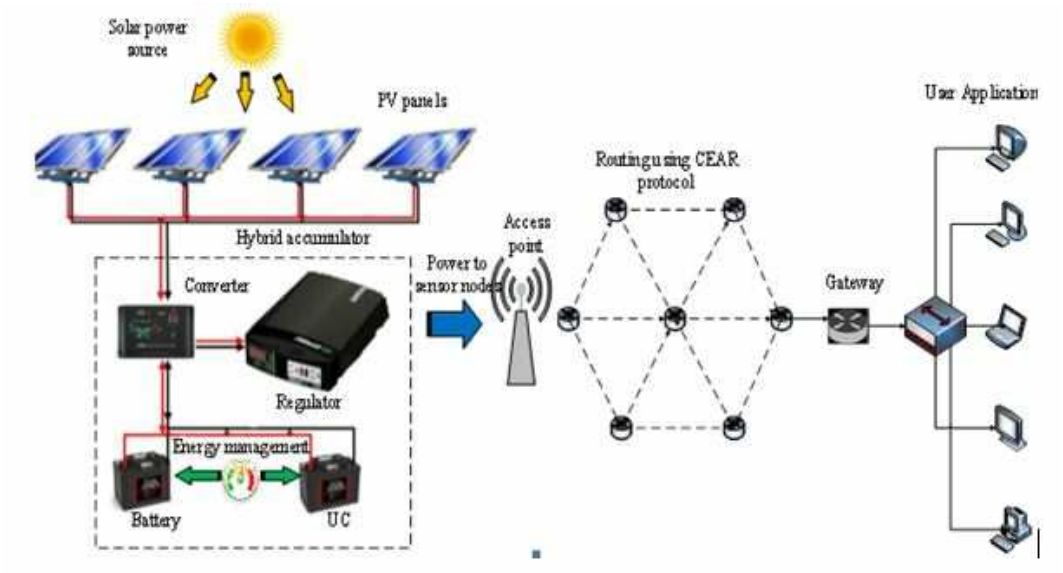


Figure 1: Architecture of our proposed method

3.1 Powering up of WSN nodes by HESS model

The HESS energy model considers the battery cycle life as well as a hybrid energy harvesting system. Also we propose a cost welfare and route source metric that takes into account the nodes' battery residual energy, harvesting rate and energy requirement to route the packet. A battery and an ultra-capacitor (UC) which is combined in the hybrid energy storage system do not have a regular connection always and also there is no direct connection between the UC and the harvesting unit. When a battery exceeds its life cycle then the node will be considered as dead. The battery is recharged when its residual energy falls below a pre-set threshold (1-D)u_B where D is "Depth of Discharge (DoD)" and u_B is the maximum capacity of the battery. Let $\zeta_k$denote the arrival time of the k-th packet request at a node. If the routing protocol selects the node, the node relays the packet. The period of activity consists of route selection and relaying of packet, in which the node does not harvest energy is assumed to be ζ_p seconds long. We define$\zeta_k{}^+$=ζ_k+ ζp. The node harvests energy during the remainder of the time slot, which ζ_h seconds long. It follows that $\zeta_{k+1} = \zeta_k{}^+ + \zeta_h$. The residual energy of battery and UC for our models is described as follows.

$$UC_{ABLE} = E^{\wedge}_{UC}(n, \zeta_k) - l(j)E(n, R(j)) > 0 \tag{1}$$

$$B_{ABLE} = E^{\wedge}_{B}(n, \zeta_k) - (1 - D)u_B - l(j)E(n, R(j)) > 0 \tag{2}$$

$$B_{RECHARGE} = E^{\wedge}_{B}(n, \zeta_k) \leq (1 - D)u_B \tag{3}$$

Where UC_{ABLE} is the time in which the UC has enough energy to route the packet. $B_{ABLE}$is the time that the battery has enough energy to route the packet.$B_{RECHARGE}$is the event that the battery has exceeded its depth of discharge and cannot accept further route requests until it has recharged. $E^{\wedge}_{UC}(n, \zeta_k)$ denotes the residual energy on the n^{th} node UC at time ξ_k, l(j) is the route for the j^{th} packet, E(.) is the energy per bit required to fulfill the route R(j), $E^{\wedge}_{B}(n, \zeta_k)$ denotes the residual energy on the n^{th} node battery at time ξ_k, and U_B denotes the maximum capacity in joules of the battery. To define the switch states S_1 and S_2 for the n^{th} node at time

ξ_k, we shall use the indicator function I(A)is 1, if the condition A is true and I(A)is 0, when A is false. We observe that S_1=1 only, if the node is selected to relay the packet. Which implies

$$S_1(n,\zeta_k) = I\{RELAY \cap (UC_{ABLE}\ cup\, B_{ABLE}\} \tag{4}$$

S_2=0 under different conditions depending on if the node is harvesting or transmitting.

$$S_2(n.,\zeta_k) = 1 - I(\overline{UC_{ABLE}} \cap B_{ABLE} \cap RELAY) - I(B_{ABLE} \cap \overline{RELAY}) \tag{5}$$

The first indicator function can be 1, if the node is chosen to relay in which case the second indicator function must be zero. The updated equations for the UC at time$_{\xi_{k-1}}$ to time $\xi_k$because of leaking, harvesting and loading is given below.

$$E^{\wedge}_{UC}(n,\zeta_k) = \min[(1-\alpha(\zeta_{k-1},\zeta_k))E_{UC}(n,\zeta^+_{k-1}) + S_2(n,\zeta^+_{k-1})\gamma_n(\zeta^+_{k-1},\zeta_k),\ u_{UC} \tag{6}$$

$$E^{\wedge}_{load,UC}(n,\zeta_k,R(j)) = l(j)E(n,R(j)S_1(n,\zeta_k)S_2(n,\zeta_k) \tag{7}$$

$$E_{UC}(n,\zeta^+_k) = \beta(n,D,\zeta_k)[E^{\wedge}_{UC}(n,\zeta_k) - E^{\wedge}_{load,UC}(n,\zeta_k,R(j))] \tag{8}$$

Where $\alpha(\zeta_{k-1},\zeta_k)$ denotes the time invariant fraction of energy leaked in the UC over a time slot, $E^{\wedge}_{load,UC}(n,\zeta_k,R(j))$ is the energy consumed by a packet if the UC is used, u_{UC} denotes the maximum capacity in joules of the UC and $\gamma_n(\zeta^+_{k-1},\zeta_k)$ denotes the energy harvested at the n-th node during time slot k-1. $E_{UC}(n,\zeta^+_k)$ denotes the residual energy on the nth node UC at ζ^+_k and $\beta(n,D,\zeta_k)$ is an indicator function for the event that the battery on node n has not exceeded its finite cycle life at the beginning of time slot k. $\beta(n,D,\zeta_k)$ is a non-increasing function of s$\xi_k$and for a fixed ξ_k, is a strictly decreasing function of D, the depth of discharge on the battery. Similarly the update equations of battery from time $_{\xi_{k-1}}$ to time ξ_k because of leaking, harvesting and loading is given below.

$$E^{\wedge}_{B}(n,\zeta_k) = \min\{[(1-\varphi)E_B(n,\zeta^+_{k-1})] + [1-S_2(n,\zeta^+_{k-1})]\gamma_n(\zeta^+_{k-1},\zeta_k)u_B\} \tag{9}$$

$$E^{\wedge}_{load,B}(n,\zeta_k,R(j) = l(j)E(n,R(j))S_1(n,\zeta_k).[1-S_2(n,\zeta_k)] \tag{10}$$

$$E_B(n,\zeta^+_k) = \beta(n,D,\zeta_k)[E^{\wedge}_B(n,\zeta_k) - E^{\wedge}{}_{load,B}(n,\zeta_k,R(j))] \tag{11}$$

Where φdenotes the time varying fraction of energy leaked in the battery in one time slot $E^{\wedge}{}_{load,B}(n,\zeta_k,R(j))$is the energy consumed by a packet if the battery is used and $E_B(n,\zeta^+_k)$ denotes the residual energy on the nthnode battery at ζ^+_k.

3.2 Routing in WSN

Routing of packets with low power consumption and low cost is a major contribution in the wireless sensor networks. In order to perform such routing, a routing protocol is needed to be designed which is cost and energy aware. In our proposed method, the major challenges in the WSNs can be implemented by the CEAR protocol. It presents an efficient and reliable routing protocol those routes the packets to the sink node. In our proposed method, the sensor nodes present in the WSN utilizes the energy supplied by the hybrid system that carries the power from the renewable source.

Setup phase

The setup phase of the CEAR protocol handles the routing table replenishment of every node. It is loaded with no less than a route to the sink node keeping in mind the end goal to route the sensing packets. This is continued with the broadcast of control packets that conveys route information concerning the sink node.

CEAR protocol

In routing, the data packets are forwarded to the hubs in the network which is a special node with an additional communication technology used to route the packets to the Base station (BS). In all sensor network applications, reliable and fast delivery of messages with reduced cost should be a major requirements. The routing performed in the WSN using our proposed CEAR routing protocol is shown in figure 2.

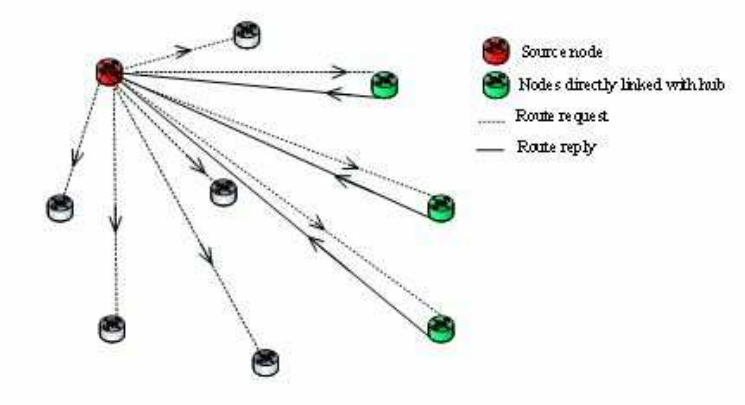


Figure 2: Routing in WSN using CEAR protocol

During data transmission, the source node transmits a Route Request (RREQ) packet to any destination hub. After the reception of packet, the destination hub transmits a Route Reply (RREP) packet to source hub. This process finds the nodes (selected nodes) that have linked with the hub which reduces the unwanted routing delay in the network which also delays the setup phase and the energy consumption. By means of this selected nodes the best path for routing is selected based on a Cost Function Request (CFR) and Cost Function Packet (CFP) which is shown in figure 3.

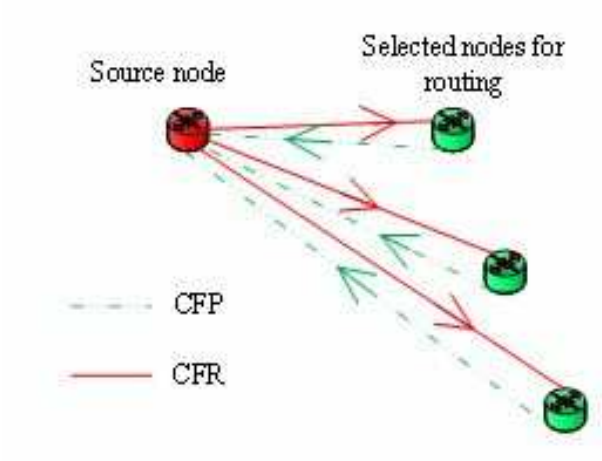


Figure 3: Path selection for routing based on CFP

To find an energy and cost efficient route from the selected nodes the source node again sends a Cost function request (CFR) to all the selected nodes. When the nodes receive the CFR, each node send a Cost function packet (CFP) which consist of the details of the cost welfare metric and the route source metric calculated for each node.

3.2.2.1 Cost welfare metric (CWM)

Cost welfare metric helps the source node to select the node with lowest cost. To ensure, an extended lifetime for the network, the cost function must reflects the node ability to forward the packet through nodes with high UC residual energy and high harvesting energy rates. Cost welfare metric based on some parameters used for routing with low cost such as hop count, transmission queue, residual energy, harvested energy, life cycle of the battery which is given by the following equation

$$\begin{aligned} CWM(n,\zeta) = (hc(n,\zeta) * w_{hc} + q_{oc}(n,\zeta) * w_{qoc}) - E_{UC}(n,\zeta) * w_{UC} \\ + E_B(n,\zeta) * w_B + \gamma(n,\zeta) * w_\gamma + L_C(n,\zeta) * W_{LC} \end{aligned} \tag{12}$$

Where hc is the hopcount which represents the minimum path length between the source node and the hub/sink node. E_{UC} and E_B represent the UC and battery residual energies in Joules. $\gamma$is the harvested energy the node scavenged in the last time slot. Lcrepresents the battery cycle lifetime. q_{oc} is the length for the node's transmission queue occupation. The hop count and transmission queue parameters are called undesirable parameters and the remaining parameters residual energy, harvested energy and battery cycle life are called as benefit parameters which supports the extension of network. The value of each parameters vary from 0 to 100. The weight function for each parameter is denoted as w_{hc},w_{qoc}, w_{UC},w_B, w_γ, w_{Lc}this allows the priority scheme in the cost metric calculation. The weight limits for each parameter is set as follows,

$$W_{hc} + W_{qoc} = 1 \tag{13}$$

$$W_{UC} + W_B + W\gamma + W_{Lc} = 1 \tag{14}$$

Where w_{hc},is the weight distribution of the hopcount,w_{qoc} is the weight distribution of the transmission queue, W_{UC} is the weight distribution of the ultra-capacitor, $w_B$is the weight distribution of the battery, $w_\gamma$is the weight distribution of the harvested energy ,$w_{Lc}$is the weight distribution of the lifecycle of battery. The weight distribution for such parameters based upon the above equations (13), (14) is given in the below table 1.

Table 1: Weight distribution for the parameters used in cost function metric

Characteristics	**Parameters**	**Weight distribution**
Positive	UC	0.45
	Battery	0.2
	Harvested energy	0.15
	Life cycle	0.2
Negative	Hop count	0.5
	Transmission queue	0.5

The battery characteristics (residual energy and cycle life) are assigned with an importance of 20%, while the harvesting energy has weight of 15%. The two "negative" (cost) parameters share responsibility of 50% of in the cost function result. WSN applications have lower harvesting energy rates that can assign higher weight to $w\gamma$, in order to customize the data flow. This formulation increases the flexibility of the cost function metric.

3.2.2.2 Route score metric

Route score metric based on energy level and weight assigned to the next hope node, link quality and weight assigned to the next hope node which is given by

$$Routescore = P_E.W_E + P_L.W_L \tag{15}$$

Where PE is the energy level of the next hope node, WE is the assigned weight for PE, PL is the link quality to the next hope node, and WL is the assigned weight for PL. The Route Score takes values from 0 to 100. Higher values indicate a better route and the packets are usually routed through nodes with higher route Score metric value. The nodes with higher route score is selected as best nodes for routing and the packets are transmitted through such nodes. The route score value of each node will be stored in the routing table and that must be refreshed at each time slot in order to choose the best node. The refreshment of route score will be done by sending a feedback packet from the neighbours with the packet length of one byte for each time slot.

Low energy and low cost routing

By the above two metric calculations CEAR routing protocol chooses the best route that consumes less power and cost. The HESS metric is zero for a node that has sufficient energy on its UC to route a packet. If the UC has insufficient energy, then a non-zero value will be calculated for HESS based on the energy state of the battery. The HESS metric depends on the cost capacity metric of, which treats battery with interminable cycle life and 100% profundity of release. Be that as it may, we have to likewise have a segment of cost connected with spending the battery cycle life; for this, we can see the cycle life of a battery correspondingly to the life of a non-rechargeable battery. Consequently, we characterize two energy depletion functions for the battery one to demoralize choice of a node that is close to its predefined profundity of release, and another to discourage utilization of a node that is close to the end of its battery cycle life. The general cost capacity ought to increment, if both of these cost segments are increased. Let us first consider the cost component for discharge within one cycle, calculate energy depletion exponent be defined as follows

$$\lambda_B(n,\zeta_k) = \frac{U_B - E_B^{\wedge}(n,\zeta_k)}{D_{u_B}} \tag{16}$$

In words, the above calculated value will be zero, when the battery is fully charged and is one, when it is discharged down to its specified level of discharge. Next, the "cost component" for the within-cycle discharge is defined as follows

$$C_{HESS}(n,\zeta_k,R(j)) = \frac{D_{u_B}}{(\gamma_n+\varepsilon)\log\mu}.(\mu^{\lambda_B(n,\zeta_k^+)}-1)l(j).E(n,R(j)) \tag{17}$$

4 Results and discussions

This section provides the simulation results and the comparison of our proposed method with the existing methods by evaluating various performance measures. The proposed method is implemented in MATLAB/Simulink platform. The proposed System configuration is given below:

Operating System: Windows 8; Processor: Intel Core i3; RAM: 4 GB; Platform: MATLAB/Simulink.

4.1 Simulation results

The simulation results obtained by our proposed method are presented in this section. Our proposed work consists of two phases which is carried out in two platforms. The simulation parameters utilized by the PV panels are given in table2 and the Specification of the other parameters used in storage system and WSN are given in table 3.

Table 2: Parameters of the PV panel

Parameters	**Specification**
Maximum power(W)	100
Maximum current(A)	5.86
Short circuit current(A)	6.44
Open circuit current(A)	21.4
Number of cells	36
Weight (kg)	12
Length(mm)	1490
Depth(mm)	35

Table 3: Specifications of the simulation parameters

Storage system Parameters	**Specification**	**WSN Parameters**	**Specification**
Converter efficiency	80%	Protocol	CEAR
UC leakage current	1 mA	Sampling time	5 us
Capacity of the battery	20%	Traffic type	CBR
Charge current	300 mA	Simulation time	600sec
Discharge current	150 mA	Number of nodes	30,50,75,100
Reference DC link voltage	300 V	Simulation area	500*500m
Nominal battery voltage	144V	Packet size	512 MB
Nominal UC voltage	125 V	Transmission range	250 m
Switching frequency	15 KHz	Node speed	20 m/s

The initial process of our research work is the extraction of energy from the solar panels and harvesting the energy to provide supply to the WSN. Our objective of proposed method is to extract the maximum amount of energy from the available solar panel and manage it by means of a storage device and then supply power to the sensor nodes. The energy obtained from the solar is based on the intensity of the light. As the intensity is high means greater voltage is obtained from the solar panel. The HEES stores the energy obtained from the solar panel and which is utilized for powering up the sensor nodes in the WSN.

The harvested solar energy is interfaced with our hybrid accumulator by the use of a converter placed between the solar panels and the accumulator. The current and voltage signals obtained by the converter are shown in figure 4(b) and (c).The energy from the harvesting unit reaches the storage unit which is the combination of both the battery and an ultra capacitor. The charging mode output of the storage unit is shown in figure 4(c).The terminal voltage obtained by the storage unit indicates both the combination of the capacitor and battery voltage. The terminal voltage obtained by the storage unit is shown in figure 4(d).SOC is normally used when discussing the current state of a battery in use. An alternate form of the same measure is the depth of discharge (DoD), the inverse of SOC. The state of charge of the battery used in our proposed architecture is shown in figure 4(e).Then the energy output from the solar panel is stored efficiently by HEES which can be further utilized for wireless sensor powering process.

The resulting power from HEES is utilized by the wireless sensor network to perform a cost and energy aware routing. The simulation is carried out in 4 different scenarios by varying the number of nodes connected in the network. The results obtained by the proposed routing protocol with different number of nodes is given the below in table 4. The simulation is carried out in 4 different scenarios by varying the number of nodes connected in the network such as 30, 50, 75 and 100 .The energy harvested from the solar panel by utilizing hybrid accumulator is utilized to power up the sensor nodes in the WSN.Inturn WSN utilizes the energy for routing

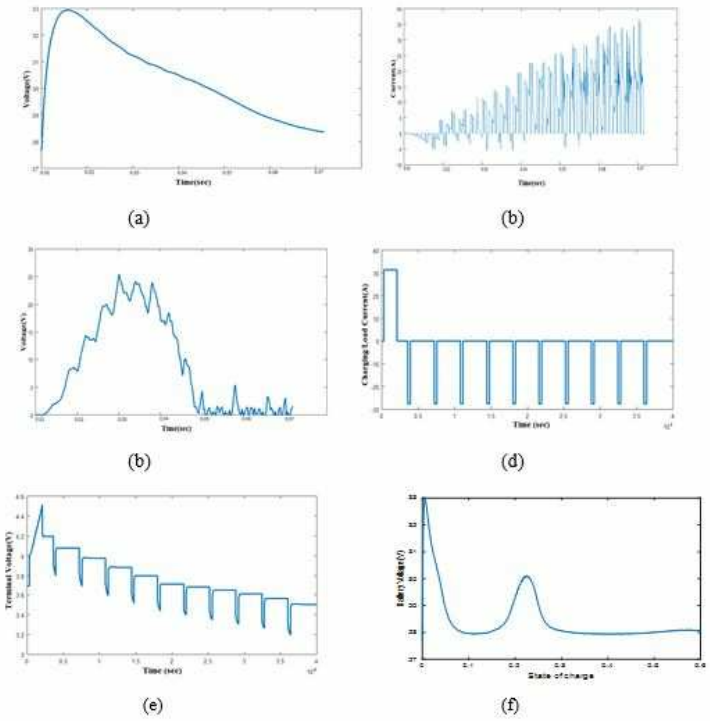


Figure 4: Energy obtained from Solar Panel,(a) Voltage obtained by the solar panel, (b) Current obtained by Boost converter, (c) Voltage obtained by boost converter (d) Charging of storage unit, (e) Terminal voltage of the storage unit, (f)State of charge of the battery

process by selecting best node from other sensor nodes for routing which is done by utilizing a cost and energy aware routing protocol. The simulation results of selecting the best node by the source node among all other sensor nodes by our proposed methodology is given below,

Thus the WSN source nodes which get supplied with the energy harvested from the solar panel is utilized for routing process which is done by utilizing the cost and energy aware routing protocol. Initially, the source node analysis with the 30 nodes based on the CEAR protocol and alternatively it searches to best nodes through the protocol along 50.75 and 100 nodes and finally chooses the best one among them and continuous the routing process. The energy is efficiently utilized by means of CEAR protocol for selecting best node for routing.

Table 4: Simulation results of CEAR protocol

Number of nodes	**30**	**50**	**75**	**100**
Packet delivery ratio	2.651	2.891	2.912	2.9867
Average E2E delay	110.442	117.981	121.157	127.982
Packet Loss	200	210	213	219
Packet Loss Ratio	0.953	1.237	1.567	1.864
Average throughput	210.225	212.872	218.902	221.086
Energy consumption	97	95	90	85

The various parameters obtained by the simulations performed in different scenario that is, with different number of nodes such as 30, 50, 75, and 100 is shown in Figure 9.

The simulation results shows that, our proposed routing protocol selects the best node from the total available nodes and forward the data with high efficiency and low cost. Above

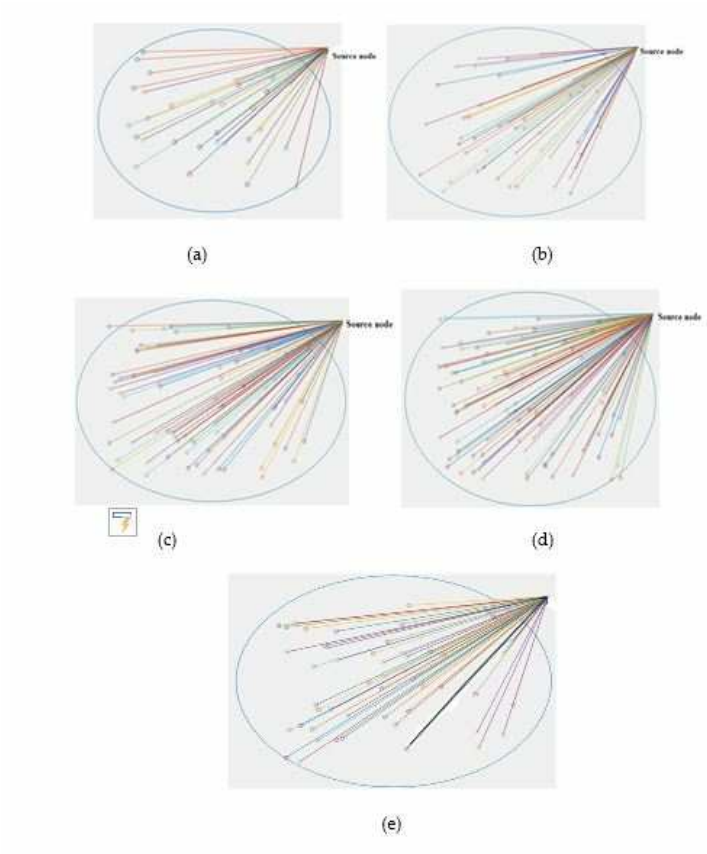


Figure 5: Simulation output for selecting best node by alternate path analysis (a) With 30 nodes, (b) With 50 nodes, (c) With 75 nodes, (d) With 100 nodes and (e) Selection of best node.

results are the evident to show the efficiency of our proposed CEAR protocol. In order to justify our proposed protocol is better than the existing protocols. The performance of the different protocols are calculated in terms of different parameters and compared in the upcoming sections.

4.2 Performance Evaluation of CEAR:

The performance of the protocol depends on various parameters and chosen for simulation. The main parameters are packet size, no of nodes, transmission range and the structure of the network. Based on the above parameters the following evaluation is calculated.

Packet delivery ratio (PDR):

The packet delivery ration is defined as the performance measure of routing protocol in any network in terms of packet delivery level. If PDR is high, then the performance is better. The relation for packet delivery ratio is as follows

$$Packet\ delivery\ ratio = \frac{\sum(Total\ packets\ received\ by\ all\ destination\ node)}{\sum(Total\ packets\ send\ by\ all\ source\ node)} \tag{18}$$

Average End-to-End Delay:

The average end-to-end delay can be obtained by computing the mean of end-to-end delay of all successfully delivered messages. If the distance between source and destination increases,

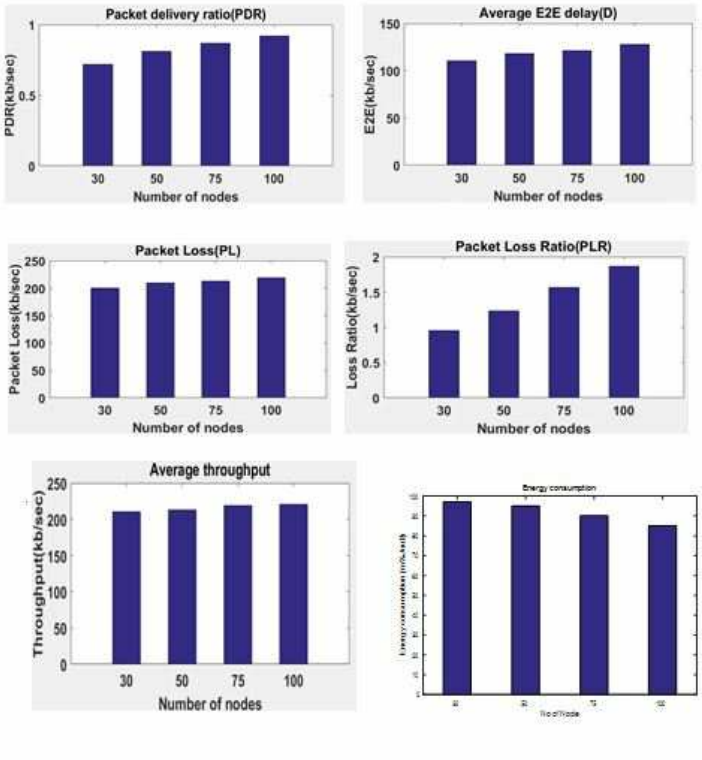


Figure 6: Simulation Results of CEAR protocol in terms of PDR, E2E delay, Packet loss, PLR and Throughput and the energy consumption.

then the probability of packet drop is also increases. The Average End-to-End Delay is expressed as follows

$$D = \frac{1}{n}\sum_{i=1}^{n}(T_{r_i} - T_{S_i}) * 1000[ms] \tag{19}$$

Where D is the average end to end delay, I is the packet identifier, n is the number of packets successfully delivered, $T_{r_i}$is the reception time and $T_{S_i}$is the send time.

Packet Loss:

Packet Loss is the ratio of the number of packets that never reached the destination to the number of packets originated by the source which are given by

$$PL = \frac{(nSentPackets - nReceivedPackets)}{nSentPackets} \tag{20}$$

Packet Loss Ratio:

Packet Loss Ratio is the percentage of ratio of the number of packets that never reached the destination to the number of packets originated by the source (PL*100) which is given by

$$PLR = \frac{(nSentPackets - nReceivedPackets)}{nSentPackets} * 100 \tag{21}$$

Average Throughput:

It is the average of the total throughput. It is also measured in packets per unit Time Interval Length (TIL).

$$Average\,throughput = \frac{Received\,size}{(Stop\,time - Start\,time)} * (\frac{8}{1000}) \tag{22}$$

4.3 Comparison Results

The use of HESS in our proposed routing method is suggested due to the better storage capacity, efficiency, energy density, power density, response time, cycle life time and cost when compared to other storage devices used in the existing methods. The importance of our proposed HESS in our proposed routing protocol is shown by the below table 3.

Table 5: Comparison of several Energy Storage Systems

	Battery	**SMES**	**Flywheel**	**SC**	**NaS**	**Hybrid ESS(proposed)**
Efficiency (%)	60-80	95-98	95	95	70	98-99.9
Energy density (Wh/Kg)	20-200	30-100	5-50	50	120	30-200
Power density (W/Kg)	25-1000	10-1000	2000	4000	120	4500
Response time (ms)	30	5	5	5	100	3
Cycle life (time)	200-2000	1000	20000	50000	2000	50000
Cost (S/kW h)	150-1300	2000	380-2500	250-350	450	100-150

The comparison graph between HESS and other existing energy storage devices regarding the above table is shown in figure 7.

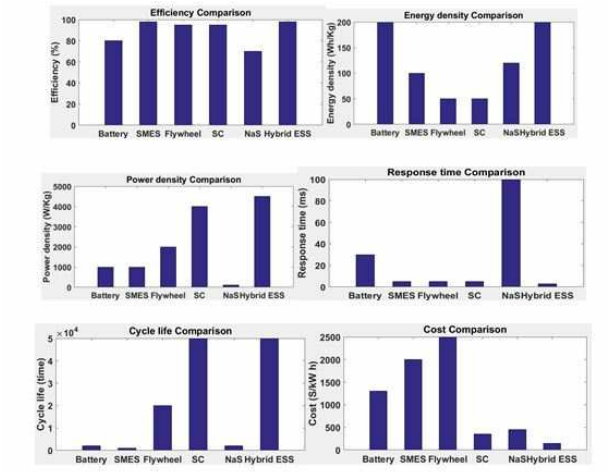


Figure 7: Comparison Results for Efficiency, Energy density, Power density, Response time, Cycle life time, Cost

The figure 7 shows the comparison graph, the efficiency of the HESS is likely to be the same when comparing to the SMES, Flywheel, and Super capacitor. But when comparing in all other factors such as energy density, power density, cycle life, response time, cost HESS is proved to be better solution. We can conclude that the use of HESS in our proposed routing provides better solution and improves the overall performance of the network. By the utilization of the HESS in our proposed routing protocol selects the path with best nodes that consumes less power and low

cost. Ad hoc On-Demand Distance Vector (AODV), Proactive Source Routing (PSR), Greedy Perimeter Stateless Routing (GPSR) are compared with our proposed routing protocols is shown below in table 6.

Table 6: Comparison of our proposed routing protocol with existing protocols

Number of Nodes	**Type of routing protocols**	**Packet Loss**	**Average E2E Delay**	**Packet Delivery Ratio**	**Average Throughput**	**Packet Loss Ratio**	**Energy consumption**
30	AODV	248	120.442	9.3671	207.864	2.890	99
	PSR	246	127.754	70.809	210.56	1.590	99.2
	GPSR	235	110.750	70.809	218.56	1.050	99.56
	Proposed	205	109.543	93.608	240.9	1.152	97
50	AODV	644	131.145	2.891	212.872	2.890	96
	PSR	173	74.7002	40.4567	214.55	1.678	97.25
	GPSR	160	70.7008	45.1786	248.55	1.150	98.32
	Proposed	210	117.981	93.6028	240.9	1.237	95
75	AODV	799	130.306	1.4598	136.936	5.999	93
	PSR	383	193.11	10.99	150.9	1.909	94.25
	GPSR	280	110.90	11.4177	190.18	1.190	95.12
	Proposed	265	102.375	98.1747	240.58	1.143	90
100	AODV	1285	129.825	1.2843	146.431	6.789	88
	PSR	313	142.524	10.99	150.9	1.909	89.21
	GPSR	280	110.90	1.2919	198.33	1.190	90.25
	Proposed	274	108.532	92.3664	266.87	1.678	89

The figures 8 to 13 shows the comparison graphs in terms of the important parameters resulted by the routing protocol for different number of nodes regarding to the above comparison tables.

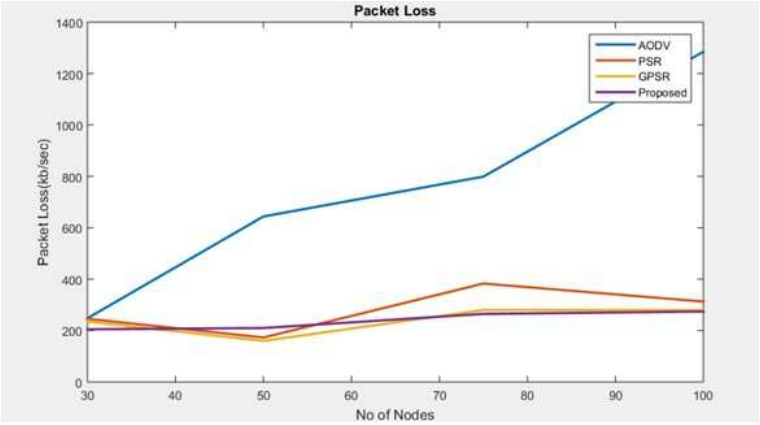


Figure 8: Packet Loss Comparison of proposed CEAR protocol with exisiting protocols

Figure 8 shows the packet loss comparison of different type of routing protocols with our proposed CEAR routing protocol with varying number of nodes. The packet loss denotes the inability of the path selected for transmission in the network. For a successive node the packet loss should be as minimum as possible. From the figure 11 it is visible that the packet loss of our proposed protocol is very less compared to other protocols. This makes us to say that our proposed protocol is best for routing in WSN.

Figure 9 shows the E2E delay comparison of different type of routing protocols with our proposed CEAR routing protocol with varying number of nodes. If the E2E delay exists in a

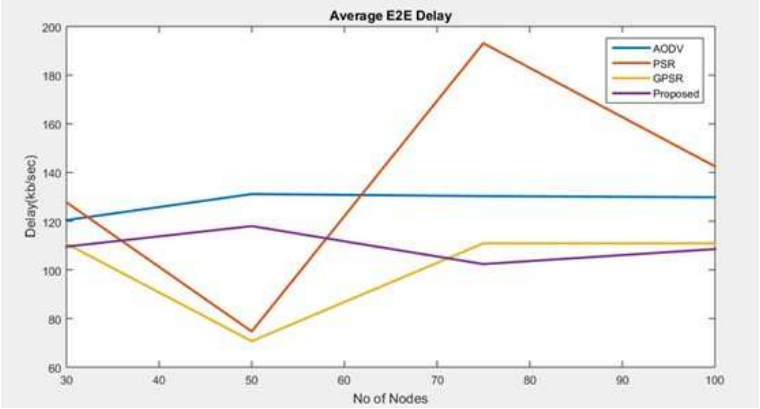


Figure 9: E2E delay Comparison of proposed CEAR protocol with exisiting protocols

network at a maximum range the performance of the overall network is degraded even though the power consumption and the cost of routing is low. From figure 12 we can demonstate that our proposed CEAR routing protocol produces minimal delay compared to other existing protocols.

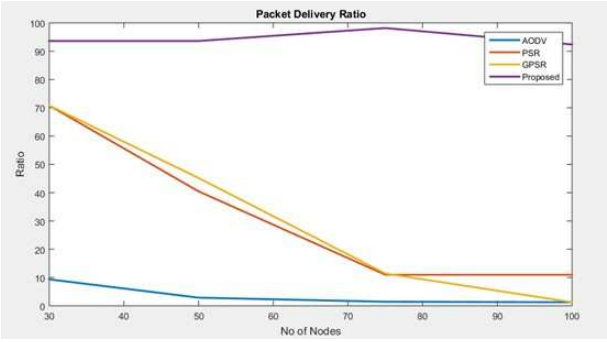


Figure 10: Packet delivery ratio Comparison of proposed CEAR protocol with exisiting protocols

Figure 10 shows the packet delivery ratio comparison of different type of routing protocols with our proposed CEAR routing protocol with varying number of nodes. Packet delivery ratio is the maximum ability of the network to deliver a packet from source to destination. From figure 13 it is evident that the packet delivery ratio of our proposed routing protocol is higher than the exisiting protocols which makes us to come a decision that we can use our proposed routing protocol to deliver maximum packets without loss in the network.

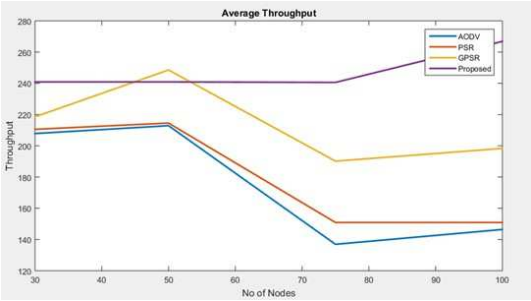


Figure 11: Throughput Comparison of proposed CEAR protocol with existing protocols

Figure 11 shows the throughput comparison of different type of routing protocols with our

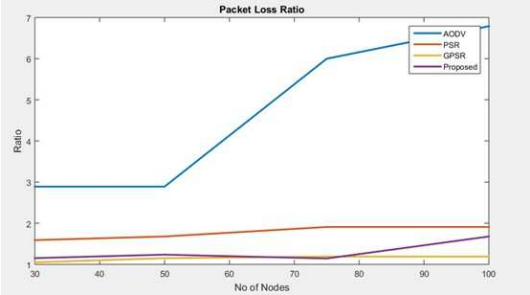


Figure 12: Packet loss Ratio Comparison of proposed CEAR protocol with exisiting protocols

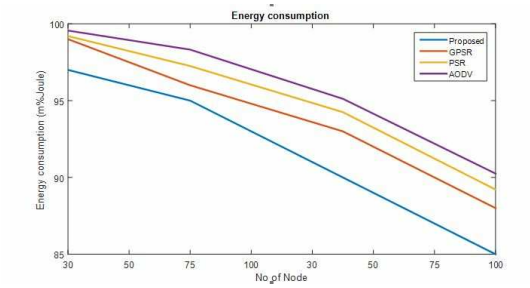


Figure 13: Energy consumption Comparison of proposed CEAR protocol with existing protocols

proposed CEAR routing protocol with varying number of nodes. For an efficient network routing protocol the throughput should be maximum as possible. From figure 11, it is shown that the proposed routing protocol results in high throughput than the existing protocols. From this we can come to know that our proposed routing protocol is efficient than the other existing protocols.

Figure 12 shows the packet loss ratio comparison of different type of routing protocols with our proposed CEAR routing protocol with varying number of nodes. Packet loss ratio indexes the variation between the number of packets send from the source to the number of packets received at the destination. The reduced packet loss ratio indicates the value of the routing protocol. It is clear that our proposed routing protocol produces less packet loss ratio which indicates that the number of delivered packets is maximum.From the figure 13 it is visible that the energy consumed by our proposed methodology is very less compared to that of other routing protocols. From the above results and comparison, it clearly shows that our proposed routing protocol selects the best node with less power consumption and low cost and also it shows that the utilization of HESS for routing by the sensor nodes in our proposed method is the most optimal solution to reduce the power consumption.

5 Conclusion

In this paper we have proposed energy and cost aware routing protocol used in wireless sensor network. The sensor nodes present in the WSN utilizes the power, which is generated by the renewable energy source such as solar energy by means of a storage system. The storage and management of the extracted energy is done by means of a hybrid storage device such as the combination of a battery and an ultra-capacitor (UC). The UC suits the primary buffer of the storage system by preventing the quick exhaustion of the battery cycle lifetime. By using a cost and energy aware routing protocol at the network level, we acquired a noticeable increment in the network residual energy without trading off the data packet delivery. Moreover in this

paper we also analysis the performance of the network while transmission by means of number of parameters. From the results and comparison such as Packet loss, end to end delay, packet delivery ratio, throughput, packet loss ratio shows that the proposed method is a technically viable option among the available other existing techniques.

Author contributions. Conflict of interest

The authors contributed equally to this work. The authors declare no conflict of interest.

Bibliography

[1] Abeywardana, D.B.W.; Hredzak, B.; Vassilios, A.G. (2015). Single-Phase Grid-Connected LiFePO 4 Battery-Supercapacitor Hybrid Energy Storage System With Interleaved Boost Inverter, *IEEE Transactions on Power Electronics*, 30(10), 5591–5604, 2015.

[2] Bajpai, P.; Dash, V. (2012). Hybrid renewable energy systems for power generation in stand-alone applications: A review, *Renewable and Sustainable Energy Reviews*, 16(5), 2926-2939, 2012.

[3] Cao, J.; Emadi, A. (2012). A new battery/ultracapacitor hybrid energy storage system for electric, hybrid, and plug-in hybrid electric vehicles, *IEEE Transactions on Power Electronics*, 27(1), 122-132, 2012.

[4] Choi, M.-E.; Kim, S.-W.; Seo, S.-W.(2012). Energy management optimization in a battery/supercapacitor hybrid energy storage system, *IEEE Trans. Smart Grid*, 3(1), 463-472, 2012.

[5] Das, D.; Roy, A.K.; Sinha, N. (2012). GA based frequency controller for solar thermal–diesel–wind hybrid energy generation/energy storage system, *International Journal of Electrical Power & Energy Systems*, 43(1), 262-279, 2012.

[6] Díaz-González, F.; Sumper, A.; Gomis-Bellmunt, O.; Villafáfila-Robles, R. (2012). A review of energy storage technologies for wind power applications, *Renewable and Sustainable Energy Reviews*, 16(4), 2154-2171, 2012.

[7] Erdinc, O.; Uzunoglu, M. (2013). Optimum design of hybrid renewable energy systems: Overview of different approaches, *Renewable and Sustainable Energy Reviews*, 16(3), 1412-1425, 2013.

[8] Etxeberria, A.; Vechiu, I.; Camblong, H.; Vinassa, J.M. (2012). Comparison of three topologies and controls of a hybrid energy storage system for microgrids, *Energy Conversion and Management*, 54(1), 113-121, 2012.

[9] Evans, A.; Strezov, V.; Evans, T.J. (2012). Assessment of utility energy storage options for increased renewable energy penetration, *Renewable and Sustainable Energy Reviews*, 16(6), 4141-4147, 2012.

[10] Glavin, M. E.; Hurley, W.G.(2012). Optimisation of a photovoltaic battery ultracapacitor hybrid energy storage system, *Solar Energy*, 86(10), 3009-3020, 2012.

[11] Guerrero, J.P.; Chiang, L.M.; Lee, T.-L.; Mukul Chandorkar, M. (2013). Advanced control architectures for intelligent microgrids—Part II: Power quality, energy storage, and AC/DC microgrids, *IEEE Transactions on Industrial Electronics*, 60(4), 1263-1270, 2013.

[12] Huang, W.; Qahouq, J.A.A. (2015). Energy sharing control scheme for state-of-charge balancing of distributed battery energy storage system, *IEEE Transactions on Industrial Electronics*, 62(5), 2764-2776, 2015.

[13] Laldin, O.; Moshirvaziri, M.; Trescases, O. (2013). Predictive algorithm for optimizing power flow in hybrid ultracapacitor/battery storage systems for light electric vehicles, *IEEE Transactions on power electronics*, 28(8), 3882-3895, 2013.

[14] Lee, H.; Byoung, Y.S.; Han, S.; Jung, S.; Park, B.; Jang, G. (2012). Compensation for the power fluctuation of the large scale wind farm using hybrid energy storage applications, *IEEE Transactions on Applied Superconductivity*, 22(3), 5701904-570197, 2012.

[15] Li, X.; Dong, H.; Lai, X. (2013). Battery energy storage station (BESS)-based smoothing control of photovoltaic (PV) and wind power generation fluctuations, *IEEE Transactions on Sustainable Energy*, 4(2), 464-473, 2013.

[16] Makarov, Y.V.; Du, P.; Kintner-Meyer, M.C.W.; Jin, C.; Illian, H.F. (2012). Sizing energy storage to accommodate high penetration of variable energy resources, *IEEE Transactions on sustainable Energy*, 3(1), 34-40, 2012.

[17] Marano, V.; Rizzo, G.; Tiano, F.A. (2012). Application of dynamic programming to the optimal management of a hybrid power plant with wind turbines, photovoltaic panels and compressed air energy storage, *Applied Energy*, 97, 849-859, 2012.

[18] Mukherjee, N.; Strickland, D.; (2015). Control of second-life hybrid battery energy storage system based on modular boost-multilevel buck converter, *IEEE Transactions on Industrial Electronics*, 62(2), 1034-1046, 2015.

[19] Nanda, A; Rath, A.K. (2018). Fuzzy A-Star Based Cost Effective Routing (FACER) in WSNs, *Progress in Advanced Computing and Intelligent Engineering*, 557-563, 2018.

[20] Onar, O.C.; Khaligh, A. (2012). A novel integrated magnetic structure based DC/DC converter for hybrid battery/ultracapacitor energy storage systems, *IEEE transactions on smart grid*, 3(1), 296-307, 2012.

[21] Padyal, R.H; Kadam, S.V. (2017). Continuous neighbour discovery approach for improvement of routing performance in WSN, *Convergence in Technology (I2CT), 2017 2nd International Conference for*, 675-679, 2017.

[22] Gupta, S.K.; Kuila, P; Jana, P.K.(2017). GA Based Energy Efficient and Balanced Routing in k-Connected Wireless Sensor Networks, *Proceedings of the First International Conference on Intelligent Computing and Communication*, 679-686, 2017.

[23] Ren, G.; Ma, G.; Cong, N. (2015). Review of electrical energy storage system for vehicular applications, *Renewable and Sustainable Energy Reviews*, 41, 225-236, 2015.

[24] Selvi, M.; Velvizhy, P.; Ganapathy, S.; Nehemiah, H.K; Kannan, A.(2017). A rule based delay constrained energy efficient routing technique for wireless sensor networks, *Cluster Computing*, 1-10, 2017.

[25] Tan, X.; Li, Q.; Wang, H. (2013). Advances and trends of energy storage technology in microgrid, *International Journal of Electrical Power & Energy Systems*, 44(1), 179-191, 2013.

[26] Xie, Q.; Wang, Y.; Kim, Y.; Pedram, M.; Chang, N. (2013). Charge allocation in hybrid electrical energy storage systems, *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 32(7), 1003-1016, 2013.

[27] Zhang, B.; Simon, R.; Aydin, H. (2013). Harvesting-aware energy management for time-critical wireless sensor networks with joint voltage and modulation scaling, *IEEE Transactions on Industrial Informatics*, 9(1), 514-526, 2013.

[28] Zhao, H.; Wu, Q.; Hu, S.; Xu, H.; Rasmussen, C.N. (2015). Review of energy storage system for wind power integration support, *Applied Energy*, 137, 545-553, 2015.

[29] Zhou, Z.; Benbouzid, M.; Charpentier, J.F.; Scuille, F.; Tang, T. (2013). A review of energy storage technologies for marine current energy systems, *Renewable and Sustainable Energy Reviews*, 18, 390-400, 2013.

INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL
ISSN 1841-9836, e-ISSN 1841-9844, 14(2), 253-271, April 2019.

The Biological as a Double Limit for Artificial Intelligence: Review and Futuristic Debate

A. Tugui, D. Danciulescu, M.-S. Subtirelu

Alexandru Tugui*
"A. I. Cuza" University Iaşi
700506 Iasi, Carol I, 11, Romania
*Corresponding author: altug@uaic.ro

Daniela Danciulescu
University of Craiova
00585 Craiova, A.I.Cuza, 13, Romania
danadanciulescu@gmail.com

Mihaela-Simona Subtirelu
University of Medicine and Pharmacy, Craiova
200349 Craiova, Petru Rares, 2, Romania
E-mail: mihaela.subtirelu@yahoo.com

Abstract: This paper aims to identify to what extent artificial intelligence (AI) is biologically limited and to launch a debate on the issue of overcoming these limitations. To achieve our goal, we utilized a qualitative research methodology framework, providing an in-depth analysis of AI limitations formulated by prominent scholars within this field of specialization. We found that the biological boundary imposes a double limitation on AI, both from a gnoseological perspective and from a technological perspective. This twofold limitation of AI underpins the idea that as long as the biological cannot be understood, formalized, and imitated, we will not be able to develop technologies that mimic it. By adopting an original approach, our research paper focused on mapping out the twofold limitation of the biological with reference to the success of AI. Special attention was paid to the motivational analysis of this limitation in terms of human existence, the opportunity and utility to create artificial intelligences as superior to the human-like condition. We have opened the door for future debates on the need to decode cellular communication by understanding and developing a *natural language of the living cell* (N2LC). Based on the present research, we proposed that within the current technological context, biological computers (biocomputing) could represent a so-called *invisible hand* outstretched by biological systems towards AI.
Keywords: Biological computer, super-AI-slices, technological singularity; limits of artificial intelligence, natural language of the living cell (N2LC).

1 Introduction

Nowadays society is dominated by technology [2, 27, 35], communications,and interaction (analog and/or digital) between different chaotic systems [25], which are sometimes too complicated [1, 38]. Change is essential [4, 7, 24, 33, 44] in our society. As we currently find ourselves at the end of the 801st Tofler lifetime [50], further predictions are being made about how society will develop in terms of technology in the following Toflerian lifetime. One of these predictions, advocated by Ray Kurzweil [30], is that in 2045, human society will reach the point where "the accelerated technological progress will overcome the human ability to understand, evaluate, and control all of its consequences, and when the non-biological intelligence created in that year will

be a billion times stronger than human intelligence today" [53]. This technological stage, mentioned by Kurzweil [30] and other field-related specialists [57], is referred to as the *technological singularity*, which translates into technological simplicity with artificial intelligence (AI).

The term technological singularity—put forward by A. Turing [18, 53, 54] in the middle of the 20th century, just before the launching of the AI concept—has played a central role in maximizing profits within various industry and business sectors. The focus on technology simplification imposed by technological singularity has been taken over by artificial intelligence that intersects with genomics and synthetic biology [45], hence placing the biological system as a key element of the future great computing platform [37]. As highlighted by Gartner [8, 12], AI has become the core of new technologies, motivating the need to identify AI limitations in order to avoid and even overcome them.

Embarking to highlight the central role of the biological system in the success of artificial intelligence systems, the present paper aims to identify to what extent the biological system stands as a technological limitation of AI.

2 Methodological issues

To reach the goal set in the present research study, we set out to identify those AI limitations governed by the nature of the imitation of the biological system (brain-body-behavior), hence formulating the following research question: *To what extent is the biological system limit for AI?* To obtain an answer to this question, we mainly utilized a qualitative approach, since throughout time various reliable approaches have already been created with regard to the limits of AI, which made our mission easier. Under the circumstances, we carried out an in-depth retrospective analysis via a meta-analysis [61] of the opinions expressed by prestigious authors within this field of specialization, i.e: Hubert L. Dreyfus, the author of *What computers can't do*; Jacob T. Schwartz, mathematician, computer scientist, former professor of computer science at New York University, creator of mathematical and computer theories, and author of Technical Report # 212 of March 1986 on AI Limits; Donald Norman, cognitive scientist; Gordon Bell, senior researcher at Microsoft and computer industry consultant; James N. Gray, specialist in database and transaction processing computer systems; Franz L. Alt, former president of the ACM; Paul W. Abrahams, consulting computer scientist and former president of the ACM; experts invited by Denning and Metcalfe [13] to put forward their views in the volume *Beyond Calculation: The Next fifty year in computing*; Max Lungarella, Fumiya Iida, Josh C. Bongard, and Rolf Pfeifer, authors of numerous AI research studies and projects and coordinators of the proceeding volume, *50 Years of Artificial Intelligence. Essays Dedicated to the 50th Anniversary of Artificial Intelligence framing the limitations of AI in the 21st century - With Historical Reflections* [34]. Figure 1 illustrates the scope of our meta-analysis for the first 50 years following the launch of the AI concept, as well as references to the predictions launched by the selected authors.

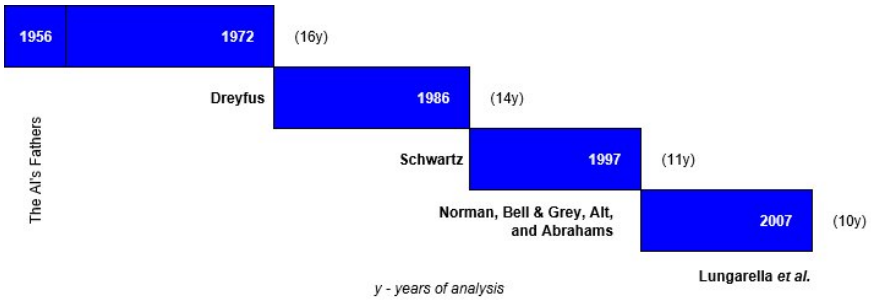


Figure 1: The meta-analysis of the timeline for AI limits

To design and develop our approach, we departed from the outline of the main AI milestones, as well as the initial goals as imagined by the visionary parents of AI. To extract those limitations imposed by the formalization and production of biological-type phenomena, we have undertaken a retrospective analysis of AI limitations as formulated by the selected authors. Having framed the above-mentioned limitations, we went a step further, highlighting their similarities to biological entities in order to establish to what extent such limitations impose actual limitations on AI. We underpinned our research enquiry with systematic implementation of a hybrid approach between biological entities and computers [41] to finally formulate some predictions with regard to what is within the reach of AI and what is not until the technological singularity is attained in our society.

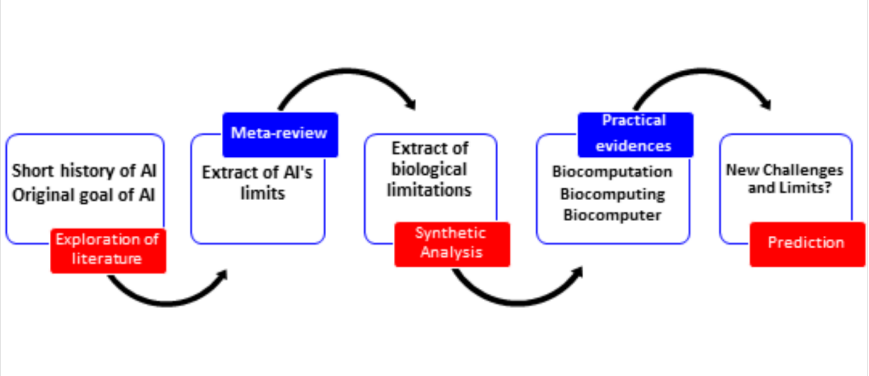


Figure 2: The research modeling stages

The overall approach of our research designed to achieve the goal set is outlined in Figure 2.

3 Artificial intelligence: a short history and its original goals

Humans have constantly resorted to technology to achieve goals with the interest of survival and controlling the others. Defined as the scientific study (logos from *λόγος* [Gk]) of craftsmanship (techne from *τέχνη* [Gk]), technology has dominated the last 10,000 years of human existence [30] and been the catalyst for the leap from one technological era to the next, including the current era of cybernetics, which was marked by the launch of the first electronic computer in 1946 (ENIAC-Electronic Numerical Integrator And Computer). In the first decade of the cyber revolution, the idea of technological intelligence emerged with the launch of the concept of artificial intelligence at the 1956 Dartmouth Conference, a concept suggested by John McCarthy to his colleagues Marvin Minsky, Allen Newell, Claude Shannon, Herbert Simon, Oliver Selfridge, and Ray Solomonoff [34]. The concept itself was the scientific response to science fiction ideas [53], which included *Elektro and Sparko* at the 1939 World's Fair; Isaac Asimov's *Three Laws of Robotics* published in the May 1941 issue of Astounding Science Fiction; the 1950 novel *I, Robot*, written by I. Asimov; and A. Turing's test in 1951 to answer the question "Can machines think?".

Newell, Shaw, and Simon's validation of 38 of the 52 Theorems of the Principia Mathematica by means of the Logic Theorist software; the 1958 launch of LISP (the first language of AI) by John McCarthy; the launch of the General Problem Solver in 1959 by Newell, Shaw, and Simon to solve complex problems, such as the *missionaries and cannibals* scenario; the publishing of the article "Pattern Recognition by Machine" by Selfridge and Neisser [47]; and the development, between 1959 and 1962, by J. McCarthy and his students at MIT of the first credible chess game

software, known as *A Chess Playing Program for the IBM 7090 Computer*, are a few concrete landmarks on the Phase I timeline of AI (1957-1962), as delineated by Dreyfus [15]. Shadowed by the highly interesting achievements carried out during Phase I, the following five years (1963-1968), framed by Dreyfus [15] as Phase II of AI, seem to be less spectacular in reaching further AI objectives such as the simulation of human behavior (particularly in relation to what the human brain can do), highlighting the fact that progress in AI has not been exciting or spectacular [51] and that the following years' predictions (after 1968) are not bright [26]. This view of limited achievements in AI is also shared by Schwartz [46], who adopts the perspective put forward by Dreyfus [15] and highlights that even after 14 years, AI faces "very limited success in particular areas, followed immediately by failure to reach the broader goals at which these initial successes seem at first to hint", motivated in particular by the technological limitations at the time of analysis.

Even 50 years after the term Artificial Intelligence was launched, at the July 2006 Conference in Monte Verita (Ascona, Switzerland), Lungarella, Iida, Bongard, and Pfeifer [34] mentioned the shared opinion of fellow researchers that AI is far from approaching the goals originally set by the first generation of AI visionaries, whereas natural intelligence is still far from being understood.

Over the last decade, and especially after 2013, an increasing number of organizational studies and reports have been registered, focusing mainly on digital economy-related industries. When accessing the Gartner.com platform, in a simple search for *artificial intelligence, AI, intelligent, automation*, and *robot* for the year 2018, we got 1358 entries, of which the content analysis would indicate that the first 427 entries (i.e. 31.44%) have one or more of the requested keywords in their headline.

Setting out to analyze the newsletters received from McKinsey & Company, we carried out an analytical study to validate the information explosion trend in terms of the usability of AI applications in the organizational field. Thus, based on 1189 newsletters received from McKinsey between 2014 and 2018, we developed our investigation for each year and set as query parameters the same keywords as in the Gartner.com analysis, i.e. in the title (subject) and summary.

Table 1 indicates the synthesis of our newsletter queries for 2014-2018 by keywords (title and/or summary), and Figure 3 illustrates the values reported in Table 1. According to our analysis, by April 26, 2017 there were no query titles (Subj.) directly querying artificial intelligence or AI. The situation changed radically after April 27, 2017, when we found that there were 42 pieces of information in which artificial intelligence and AI appeared both in the title and the summary of the investigated information. It is important to note that a McKinsey newsletter can include from one to ten independent notices in which keywords appear either in the summary or in the title and summary. Considering only the newsletter content, we found that after April 27, 2017, the use of the terms selected either in the title or in the title and summary increased from 32 entries in 2014-2017$_a$ to 237 entries between 2017$_a$ -2018, which means an increase of 877% after April 26, 2017.

Consequently, for both the Gartner and McKinsey data processed in relation to the reports, research studies, and surveys about technological trends at the societal level, we marked out an industry-oriented expansion of applied AI theoretical concepts to solve practical issues in order to simplify problem solving and/or replace individuals from various processes. In addition, our organizational analysis registered the occurrence of an inflection point on April 27, 2017, at which time it can be seen that the McKinsey reports focused on the technological expansion of intelligent applications in different industries. This is in fact another argument, an obvious proof of the social view in favor of complexity simplification via the instruments of artificial intelligence on the way towards the technological singularity.

Table 1: Synthesis of McKinsey Queries

Total	**2014**	**2015**	**2016**	**2017$_a$**	**2017**	**2018**
Artificial Intelligence $_{(KW)}$	**7**	**6**	**0**	**0**	**21**	**48**
Artificial intelligence (Subj.)	0	0	0	0	5	11
AI $_{(KW)}$	**0**	**0**	**0**	**0**	**21**	**56**
AI: (Subj.)	0	0	0	0	2	16
Robot $_{(KW)}$	**2**	**0**	**1**	**2**	**9**	**6**
Robot: (Subj.)	1	0	1	0	1	2
Automation $_{(KW)}$	**2**	**1**	**1**	**3**	**31**	**33**
Automation: (Subj.)	2	1	1	3	10	11
Intelligent $_{(KW)}$	**0**	**0**	**0**	**2**	**3**	**4**
Intelligent: (Subj.)	0	0	0	1	1	1

Legend: KW=Key words (Subject+Summary); Subj.=Subject; 2017$_a$: until April 26 2017.

4 The limits of AI in time — a meta-analysis

According to economic theory, every technological revolution is emphasized by the emergence of a new production factor that triggers the concrete manifestation of the new technological age. Under the circumstances, Tapscott [49] has labeled the current economy the digital economy, stating the significant contribution of information to the creation of GDP (Gross Domestic Product) as a specific factor in the cybernetics era [59]. From a societal perspective, Tapscott records an accelerated transition of human society from the industrial society of the early twentieth century to the Internet-dominated post-industrial society and digital technologies specific to the late twentieth century.

In terms of artificial intelligence as a digital technology of the 21st century, our research study aimed to pinpoint the main landmarks as highlighted by the AI visionary parents from its very beginning and analyze to what extent these limitations have been preserved over time and how they influence the evolution of this field of specialization.

In essence, AI was seen as an attempt to build systems with human-like or even super-human capacities in certain domains [60], traditionally considered closely related to the human mind. In this endeavor, in order to better understand their target, the pioneers of AI were keen to discover how the human brain functions. Originally considered a *meat machine* by M. Minsky in the late 1960s, two decades later the human brain was characterized by Schwartz [46] as a *biochemical computer*. Without actually defining the human brain, P.W. Abrahams [1], a former student of M. Minsky at MIT, postulated in his essay "The World Without Work" that the human brain is a target that is hard to understand and mimic using artificial intelligence. The author compares the human brain to the Moon towards which the Earthlings have set out to build a tower, but no matter how hard they work to raise it, it is still not enough compared to the Earth-Moon distance.

4.1 AI limitations according to Hubert L. Dreyfus

In his 1972 work entitled *What Computers Can't Do*, Hubert L. Dreyfus carried out a critical analysis of what AI managed to do or not with regard to the expectations postulated by the AI visionary parents during 1956-1972. As illustrated in Figure 1, Dreyfus's [15] answer to the book title question, *What Computers Can't Do*, formulated from a critical perspective on AI, explicitly mentions in the conclusion the limits of artificial intelligence.

Dreyfus [15] explained what computers cannot yet do due to technological limits even with

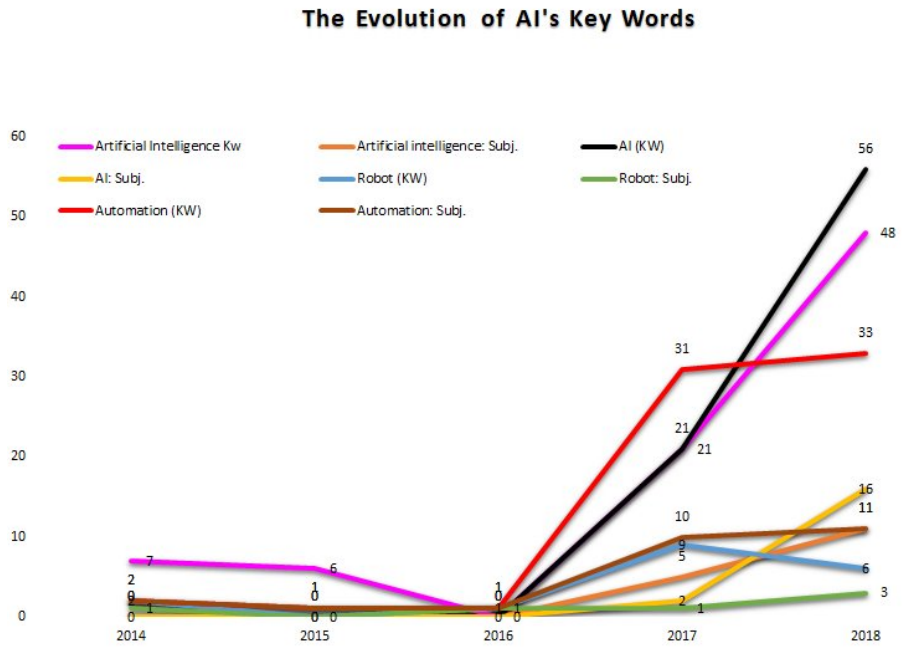


Figure 3: A graphical representation of the organizational explosion of the AI keywords

the creation of equipment capable of huge performances with up to $10^{10^{10}}$ states (Dreyfus's number). He referred to the limitations of processes of informational formalization in the brain and the body, and, additionally, the limits of human behavior formalization (for which there are sometimes no rules), as well as the limitation of the formalization of the non-material aspect of the human soul (immaterial soul) inspired by Descartes' [14] vision and the motives of his Discourses.

In fact, irrespective of the research hypotheses (tested and untested) formulated by AI titans such as Minsky, Shannon, Simon, Shaw, Turing, Neumann, McCarthy, Fodor, and Feigenbaum focused on and connected to ideas and theories of interconnected research areas such as mathematics, physics, chemistry, psychology, and philosophy, AI limitations in Dreyfus's opinion were centered on two key words: technology and formalization. Dreyfus regarded formalization as the main limitation, in the sense of the impossibility of heuristic modeling from the biological, psychological, ontological, and epistemological perspective of the *brain-body-behavior-soul* (SoBrBoBe) grouping in relation to human needs optimization functions.

4.2 AI limitations according to J.T. Schwartz

A state-of-the-art approach to AI limitations was developed by J.T. Schwartz [46], former computer science professor at New York University. Schwartz analyzed what had been achieved in the field of AI in the 30 years after the concept release (1956-1986), as illustrated in Figure 1.

Schwartz featured two categories of limitations, namely the limitations imposed by the concrete physical and logical issues of artificial intelligences design and those required by the ethical dimension of their existence. In the first category, Schwartz included a) the fundamental limits to the constructability of artificial intelligences (AIs), which refers to the (limited) possibilities of systems designed similarly to the human brain (Br) in performance; b) limits imposed by the quantitative theory of computational complexity, motivated by the remarkable, but complicated to simulate, ability of the human brain to manipulate complexity and to reveal it in a simple and

efficient manner; and c) knowledge-based limitations on AI detailed on three levels: sensors, with reference to the analysis of images (computer vision) and the analysis of nature; motor control, modeling of spatial environments, and motion planning; and reasoning, planning, knowledge representation, and expert systems with reference to graphical searching, predicate systems, expert systems, knowledge representation, and learning. In the second category of ethical limits, Schwartz included the *fear* induced when such systems are out of control, as well as the methods and rules of human interaction with these systems.

4.3 AI limitations upon the 50th anniversary of the foundation of the ACM

The debates launched in March 1997 on the occasion of the 50th anniversary of the Association for Computing Machinery (ACM) via 20 essays authored by IT experts and specialists [13] triggered an increased awareness of the technological limitations of artificial intelligence, highlighted in a direct or indirect manner by G. Bell, J.N. Gray, D. Norman, F.L. Alt, and P.W. Abrahams in their works. In fact, these limitations are found in the first section, "Coming Revolution," as well as the second section, "Computers and Human Identity," of the volume edited by P.J. Denning (chair of the Computer Science Department in the School of Information Technology and Engineering at George Mason University) and R. Metcalfe (the inventor of Ethernet technology), i.e. the debates on the predictions launched with reference to the future of computing.

Bell G. and J.N. Gray: "The Revolution Yet to Happen"

In their chapter "The Revolution, Yet to Happen," G. Bell and J.N. Gray [5] discussed cyberspace, where information about all real-world physical objects will be found online by encapsulating it in a chip, leading to fully networked systems. The authors' prediction for 2047 was that the operating and storage performance of the computer would equal the human brain (Br) and that the so-called on-a-chip systems, body area networks, and robots would make their presence felt in cyberspace.

Essentially, Bell and Gray were of the opinion that there was a technological limitation to the development (understanding, formalizing, and building) of systems capable of human-brain-like performance that could be overcome by 2047. However, hybrid systems were considered as immediate solutions (after 2025), in which body area networks (as an intermediary step towards biological computers or biocomputing) would play a considerable role.

Norman D.: "Why It's Good That Computers Don't Work Like the Brain"

"Why It's Good That Computers Don't Work Like the Brain" is the chapter [41] where psychologist Donald Norman (keen on both human and computer behavior) adopted a positive approach to the differences between the individual, as an intelligent, unpredictable, robust, relatively error-insensitive, and redundant being, and the computer (including robots) as an abstract, linear, consistent, rational, and precise machine. This mirror characterization reflects the irrefutable limits of artificial intelligence. Norman is of the opinion that computers and robots will never come to mimic or surpass people, and that due to technology, the human species is condemned to an ever-growing complexity, which will lead to a continued loss of privacy and freedom of action. Within this context, the technological limitation of computers (including artificial intelligence) compared to the human brain is decided from the concept, design, and realization stages, since we place into discussion two totally different entities, namely the human brain (Br)— the result of an evolution marked by continuous adaptations and interactions over millions of years, where the natural selection criterion was that of survival of the species — and computer technology, which is limited in terms of its the evolution over time (little over 50

years at the date of the author's analysis) as well as its design and development by reference to efficiency optimization functions and computational algorithms.

Norman was categorical and clearly stated that a computer would not be able to mimic or surpass people. The solution recommended by the author to overcome this technological limitation envisages the human-computer relationship seen as an interaction between cooperative and even hybrid systems towards the development of the so-called *biological computer*.

Alt F.L.: "End-Running Human Intelligence"

The marked difference between human intelligence and artificial intelligence is also shared by F.L. Alt [3] in his chapter "End-Running Human Intelligence." The author features several domains where AI las proven to be less successful, such as "chess playing, legal problems, medical diagnosis, weather prediction, public opinion surveys, and the understanding of natural language."

Some limitations identified by Alt are already outdated at present, proving that AI has made remarkable progress over the past two decades.

Abrahams P.W.: "A World Without Work"

Disappointed with the failure to meet the goals he had predicted during AI's debut in the late 1950s and early 1960s when he was M. Minsky's student at MIT, 40 years later, P.W. Abrahams [1], in his chapter "A World Without Work," outlined some of the most important AI failures, including not having met the goal set by Japan in the Fifth Generation Computer Project, despite the contemporary appreciations of Feigenbaum and McCorduck back in 1983 [22], and the cultural and emotional limitations of robots' interactions with human subjects in the field of service provision (such as telephone services, taxi services, and cuisine). For example, with regard to the interaction with human subjects, Abrahams added one of the most challenging limits, i.e. the inability to feel the same type of love for a robot as for a person, or even the same intensity of sexual attraction (even if 20 years later this limitation seems to be somewhat conceptually overcome via conferences on topics such *Love and Sex with Robots, LSR 2016* and the special issue "Love and Sex with Robot" in the *Robotics* journal [62]). Towards the end of his chapter, Abrahams launched a series of rhetorical questions to highlight AI behavioral limitations in comparison with the biological system (BrBoBe), as well as various aspects with regard to the utility of intelligent humanoid machine design. Among such rhetorical questions, we identified those related to the capability of intelligent humanoid machines to engage in activities or situations specific to humans, such as eating, bleeding, dying, procreating, and the feeling of pleasure or pain. Moving a step towards an affirmative answer related to procreation, Abrahams wondered if robot children would be subject to the same ethical imperatives specific to our children? Could the robots be treated like slaves without any compassion? And what would then be the reason for designing robots, apart from intellectual curiosity and desire for power, provided that humans can easily create people and do so with pleasure? We considered that all these questions, which are difficult to answer, impose the limitations of AI as a form of manifestation and level of development. In addition, Abrahams also drew societal boundaries (including ethical ones) in the sense that robots cannot succeed in creating a better world than the one people created on their own, even if robots undertook all social/economic tasks and overcame different cultural, racial, and religious differences. Consequently, the role assigned to intelligent computers is to complement the individual in his/her actions, such as managing, exploiting, recovering, and recycling the planet's limited resources.

Abrahams' approach encompasses both constructivist limitations, which involve the understanding-formalization-construction process, and behavioral limitations (including feelings, states, and manifestations specific to humans) as well ethical ones. Abrahams' recommendation

was to create systems assigned to humans (living systems) similar to those put forward by D. Norman in his chapter.

4.4 AI limitations on the 50th anniversary of the concept launch

On July 9-14, 2006, AI specialists attending the 50th Anniversary Summit of AI, held at Centro Stefano Franscini, Monte Verita, Ascona, Switzerland, celebrated 50 years since the term artificial intelligence was launched. Although the summit agenda listed the launch of speculations about the future of AI, we found that no actual limits were discussed, but rather the challenges and development directions applicable in different areas. Lungarella, Iida, Bongard, and Pfeifer [34] wrote "AI in the 21st Century - With Historical Reflections", as forward-thinking paper showing the evident advantage to having obtained a clear picture of the stage reached by AI in the first 50 years since its launch, and to having synthesized in a prudent manner expectation forecasts for the next 25-50 years. We shared the same motivation to have selected this paper for our meta-analysis timeline.

In the first decade of the 21st century, Lungarella et al. [34] stated that there was still a technological limitation imposed by the fact that *natural intelligence* (the topic of behavioral emulation by artificial intelligences) was "far from being understood", saying that the "basic theories of natural intelligence are lacking" while "artificial forms of intelligence are still much more primitive than natural ones". Thus, the authors insisted on the limitation of the understanding and the conceptual formalization of the biological system (the brain-body-environment triplet) within the constrained limitations still imposed by a rudimentary and unavailable technology. This overview of conceptual and technological limitations was articulated amid the paradigm shift in the stated purpose of artificial intelligence. Thus, if intelligence was initially thought to be located in a box in the human brain (Br), after 2000, a novel perspective focused on the distributed intelligence located throughout the whole organism (Bo), which interacts with and explores the environment (En). In other words, the authors endorsed the paradigm shift from a computational approach to an embodied perspective.

5 The result is a biological limitation

Following our meta-analysis of the main limitation-related views put forward by Dreyfus, Schwartz, Norman, Bell, Gray, Abrahams, Lungarella, Iida, Bongard, and Pfeifer, we synthesized a series of methodological assertions to highlight the essence of AI limitations.

The first methodological assertion, based on the limitations of AI as formulated by Dreyfus, states the existence of a technological limitation to building systems with huge performances ($10^{10^{10}}$ states, a value that even today does not work), which is a performance with which the biological system is indirectly credited via the brain-body binomial (BrBo), complemented by the individual's inability to achieve complete self-understanding and the formalization of his/herbehavior (Be) and soul (So) as a manifestation of the whole SoBrBoBe.

Departing from the AI limitations as formulated by Schwartz, we extracted the second methodological assertion, namely the existence of a concomitant limitation of scientific and technological knowledge in constructing artificial intelligences similar to the human brain (Br) in terms of its remarkable ability to manipulate and reveal complexity in various AI applied domains, to which ethical limitations of the biological interaction (BrBoBe) are added. We noted that Schwartz focused on the biological component (BrBo) of AI limitations, which cannot yet be understood or formalized so as to design AIs.

The experts summoned by Dening and Metcalfe [13] as contributors to the anniversary volume of the 50th anniversary of the foundation of the ACM clearly identified a series of in-

teresting AI limitations, from which we extracted the third methodological assertion to identify the technological limitation in system designs (computers) that mimic or surpass people [41] in their behavior regarding survival, feelings, moods, and manifestations within a lack of utility context [1]. In essence, the proper design involves the understanding and formalization of the biological system (BrBoBe), and the authors endorse this as a possible solution to the design of cooperative systems based on human-computer [1] interaction, hybrid body area networks [5], and the development of the biological computer [41].

The fourth methodological assertion, extracted from the AI limitations identified by Lungarella, Iida, Bongard, and Pfeifer [34], consists of the lack of basic theories with regard to what natural intelligence is as the subject of AI emulation and the continued existence of a limitation of the understanding and formalization of the biological system (BrBo) in the interaction with and exploration of the environment (En). Essentially, it is a reformulation of human behavior (Be) with the interaction with and biological exploration (BrBo) of the environment (En) that forms the BrBoEn triplet, while the biological system (BrBo) is still largely unknown.

By examining the four methodological assertions, we found that the biological represents a common and important limitation of AI, hence it must be first understood, then formalized, and finally imitated by technology. As long as the biological cannot be understood, formalized, and imitated, we will not be able to develop technologies that can mimic it. Surface imitation of natural intelligence can only lead to superficial results, such as the tower that the earthlings would propose to build in order to reach the Moon, and no matter how much of it would be built each day, it would remain insignificant compared to the Earth-Moon distance [1]. Subsequently, we can say that both the gnoseological and technological boundaries of human beings are automatically limitations of AI.

All previously mentioned authors investigated the key role of the biological component in AI limitations by limiting the initial understanding of natural intelligence — in fact, self-understanding, which must be the premise of success in the theorization and formalization of natural intelligence and the technological stage in the process of AIs design that mimics the biological system. In other words, in compliance with the four methodological assertions, all the limitations identified by Dreyfus, Schwartz, Norman, Bell, Gray, Abrahams, Lungarella, Iida, Bongard, and Pfeifer are located on the following logical trace: biological $->$ technology $->$ biological $->$ AI (Bio-Tech-Bio-AI).

We can then explain the logical route Bio-Tech-Bio-AI in the simplest way; the individual is not capable of self-understanding (*the first biological limit*) in order to create fundamental theories of natural intelligence and is still limited in the design and use of appropriate technologies (tools and materials), technologies that in turn would be used to mimic the biological system (*the second biological limit*) in order to design what we should have understood, i.e. artificial intelligences.

Consequently, our meta-analysis, based on the four methodological assertions, highlights that the biological boundary remains the main limitation of AI, both from a gnoseological perspective (due to the impossibility of self-knowledge) and from a technological perspective (given by the impossibility of formalization, design, and use of technologies — instruments and materials — that imitate natural intelligence). When implying the limitation of the biological systems, we found that the natural intelligence of the individual is limited from a gnoseological perspective. The gnoseological and technological limitations fall into a spiral pattern similar to DNA, in which the two boundaries communicate, but as of yet do not intersect. The above analysis justifies the conclusion that the human being is still at this moment a double limit for artificial intelligence.

6 Current AI records and limitations — futuristic debate

The two limitations — gnoseological and technological — approached as a double limit of the biological in terms of the success of AI, were analyzed in a direct manner by Norman [41]. In the attempt to answer "Why It's Good That Computers Don't Work Like the Brain", the author asserted that computers and robots would never come to mimic or surpass people. Accordingly, Norman advocated the idea of parallelism between the two boundaries, while he identified a bridge between them through biological computation via a biological computer. Even though the Merriam-Webster Dictionary notes that the term biological computer (bio computer) was first used in 1952 in the sense of a "computer that uses components of biological origin (such molecules of DNA) instead of electrical components", this sense was taken over in 1958 and Heinz von Foerster [55, 58] created the Biological Computer Laboratory at the Department of Electrical Engineering, the University of Illinois. It was not until 1997 that Norman endorsed a fusion-oriented approach (in the sense of merge) between the biological cell and the classical computer that would lead in the coming years to a hybrid biological computer capable of notable performance in the applicative field of artificial intelligence. Within the framework of the double biological limitation highlighted above, corroborated with societal expectations of AI design that would predominantly replace human activities, we considered it appropriate to continue our discussion with some promising achievements in recent years as arguments for the success of AI and launch further questions about future challenges and limitations.

6.1 Some actual evidence of AI

The idea of the biological cell's assimilation into the future biological computer gives us biocomputing, a concept used since 1965 [36] as an "application of computer science to biological research", though in tandem with the notion of biological computation (not yet defined in Merriam-Webster's collegiate dictionary [36]), meaning that a system of neurons, grown biologically [41] is used to design AIs. If at first the biological computer was defined as a system of neurons, artificially grown, capable of problem solving via biologically real, brain-like operations [13], later on we could speak of DNA computers [29, 45] or computers using synthetic biological components in order to manipulate, store, and retrieve data. As in any technological field, the field of biological computers is mainly concerned with achieving a stable and reliable technology. In terms of the understanding and formalization of the biological system and ultimately the development of stable and reliable technologies leading to the development biological computers, it is worth highlighting some of the most important achievements of the past five years.

Parallel computation between computers and molecular motors

In the attempt to concretize Norman's idea of efficiently involving the biological system in the creation of hybrid systems [41], the complex research team coordinated by Professor Nicolau of McGill University (Canada), with representatives from the Lund University (Sweden), Molecular Sense Ltd. (UK), Technische Universitat Dresden (Germany), Philips Research and Philips Innovation Services (The Netherlands), and Linnaeus University (Sweden) [39], obtained in 2016 the "proof of concept" of a computer that operates in parallel using mobile molecular proteins that exploit a nanotechnology-based network and codes a problem still unsolvable by electronic computers. In particular, the authors [39] put forward a "parallel-computation approach, which is based on encoding combinatorial problems into the geometry of a physical network of lithographically defined channels, followed by exploration of the network in a parallel fashion using a large number of independent agents, with very high energy efficiency." This project, initiated

by D.V. Nicolau [39], a professor at McGill University, incorporates the development of various ideas on the design of molecular motors at the micro- and nano-biocomputation level, published in 2006 in Microelectronic Engineering [40].

Stable storage of digital data in DNA

A practical achievement of great importance in the field of bio computers was the research study completed in 2016 by Erlich and Zielinski [19], researchers at Columbia University and the New York Genome Center, who informed the scientific community about a major improvement in DNA information storage and retrieval when they succeeded in implementing "a new coding strategy to encode text, images, a movie, and an operating system in 2 megabytes of DNA, and retrieve it back perfectly in multiple trials" [21]. The two researchers, through their strategy, managed to store 215 petabytes of data on a single gram of DNA [11].

Biological Computers Inside Living Cells

One of the most striking achievements in the field of bio computers was announced by A. Green, an engineer at Arizona State University, who informed the public that together with researchers from Harvard University, they had developed a biological computer "that controls how cells behave," [28, para. 1] including the construction of biological circuits that behaved similarly to digital circuits and used the logical operators AND, OR, and NOT to make decisions. The research team thus managed to create a biological transistor (called a transcriptor), a specialized computer that could be programmed to monitor and affect the functions of living cells. To achieve this, the team used DNA that could store up to 455 exabytes of data per gram [28].

Biology: the next great computing platform

Within the last two decades, and particularly after 2012, biology has enjoyed increased attention from researchers in different areas of specialization. For example, a step forward was made in the formulation of theories such as the theory of electrodynamic instabilities in biological cells [32] and the design of specialized sensors for the reception of emotions-on-a-chip [31], towards the recording of the extracellular neural activity [20], not to mention the continuous-flow biochips [43], the design of revolutionary gene-editing technologies such as CRISPRs [37], and the design of nano-biologic computers as reliable alternatives for quantum computers [9, 39]. All these intersections between AI, genomics, and synthetic biology as highlighted by Rosso [45] will help specialists turn biology into the next great computing platform [37] of society.

6.2 Futuristic debate and next questions

The topic of AI limitations has always been a sensitive issue for society in general and for the individual in particular. Following our meta-analysis, we highlighted the importance of ethical [46] and spiritual or soul-related [15] limitations that are difficult to overcome [48]. To successfully outline AI limitations, since the individual stands as the double limit meant to secure the success of AI, motivates us to shed some light on our limits as a species in the universe. In what follows, we discuss some central topics of reflection on our human-like interaction with AI.

Do we really know our limitations as a species? It is quite obvious that the individual, as a human, manifests a rather limited understanding and self-understanding. For example, from an existential perspective, we do not yet know with utmost certainty where we come from and where we are going on the path of our existence. We cannot yet understand ourselves or explain and formalize our own intelligence in a credible way, and we do not have yet an explanation of why humanity displays (at least for now) superior intelligence to the other species on Earth. We certainly do not know, and there is no clear evidence of our appearance on Earth and which of

the creationist or evolutionary theories is true [10, 16, 17, 42]. In addition, even if we were to be given a proof to demonstrate one of the two theories, we would still wonder how this evidence should be exhibited to be irrefutably credible. Are there temporal, spatial or other limits to our ability to understand that proof? On this level, the list of questions remains open. The only certainty is that we have many limitations as a species-that is, we know very little about what we want to know.

Is it necessary and appropriate to plan the design of such AI that go beyond the human being? The usefulness of the descriptive truism formulated in the previous question, rounded by the obvious conclusion that we have many limitations, helps us to clearly understand another aspect: *Someone* (from the creationist perspective) or *Something* (from the evolutionary perspective) has contributed to our development as a species. Are we able to compete with the *Creator* or the *Millennia-long evolutionary process* in our attempt to design AIs? Should we attempt to do so? And if yes, are we firmly convinced that we will succeed? And if we succeed, is it prudent for the human species that these AIs be superior to us in terms of intelligence? It is obvious that the human has been overcome in many areas by technology, as far as efficiency is concerned [56]. Is it necessary and opportune that we be overcome by technology and from intelligence perspective? The positive answer to such a question, correlated with the ethical limitation regarding the exploitation of these AI as highlighted by Abrahams [1], should encourage a serious reflection on the future sharing of the exploited and exploiter roles.

How far should AI reach? The answer is simple. As long as the individual has his/her own gnoseologic and technological limits to which the existential ones are added, AI cannot overcome us for a very simple reason—we cannot build something we can not formalize, to which we add our fears and/or pride in relation to a technological entity created by ourselves. However, biocomputing, bio computers, and DNA computers could lead to the design of AIs that overcome natural intelligence on certain levels, which means we could talk about the *slices of super AI* (super-AI-slices), without the scope to completely overcome the natural intelligence.

Biocomputing—The invisible hand of AI? Fascinated by the secrets of medicine, in an informal discussion in 2014, we asked the famous surgeon I. Lascar, a professor at the University of Medicine and Pharmacy in Bucharest, *what the secret was to a successful operation.* Among the syntheses and content-related explanations, Professor Lascar pointed out that surgery is assisted, besides a number of strictly scientific factors, by a so-called *invisible hand* that contributes to the success of an operation and which all physicians rely on. In this context, the success of biocomputing research and development as part of the bio computer could be the catalyst for leaping to a level of AI that surprises us in terms of intelligent performance and behavior. Current achievements, such as the design of the biological transducer; the monitoring, programming, and behavioral control of the live cell (via logical operations AND, OR, and NOT); and technological challenges such as the decoding of live cell communication and the future development of a *natural language of living cells* (N2LC) used in biocomputing could turn biocomputing into the invisible hand of biological systems stretched towards artificial systems, especially AI.

All the questions raised in our present paper are aspects of AI boundaries in relation to natural intelligence, but also future research topics in relation to the following premise: nature is very simple and efficient in everything she makes [52]. It is very important for us, as humans, to understand the simplicity of nature in creating biological entities, decoding the biology, and applying it to communicate via an NLLC to build calm technologies [6, 23] at the societal level.

7 Conclusion

The research study carried out in this paper, motivated by the need to clarify to what extent technological limitations are based on biological limitations and departing from the methodolog-

ical assertions extracted via our meta-analysis, highlighted that the biological boundary represents the main limitation of AI from both a gnoseological and a technological perspective. Thus, we have come to the conclusion that as long as the biological system cannot be understood, formalized, and imitated, we will not be able to develop technologies that can mimic it.

We highlighted the double biological limitation as a conclusion of the Bio-Tech-Bio-AI logical pathway, to which we assigned the results of our meta-analysis with respect to AI limitations as identified in Dreyfus, Schwartz, Norman, Bell, Gray, Abrahams, Lungarella, Iida, Bongard, and Pfeifer. We registered on the Bio-Tech-Bio-AI path the inability of the individual regarding self-understanding *the first biological limitation* in order to develop theories about natural intelligence, which induces a limitation on the creation and use of technologies appropriate to the process of imitation/mimetization of the biological system (the second biological limitation) in order to design what we should have understood, i.e. artificial intelligences. Within this framework, we conceptually embraced the gnoseological and technological limitations in a DNA spiral model through which the two boundaries communicate, but which do not intersect as of yet.

At this stage of human scientific knowledge, as far as the double limitation of AI is concerned, we highlight that the technological limit is not a mere consequence of the gnoseological limit, motivated by the fact that from a causal perspective we are talking about the same biological system which is analyzed at two different moments. At the first moment, the biological system searches for and does not find the answer to the question "What do we imitate?", while, at a second moment, the same biological system is looking for the answer to "With what (technology) do we imitate?". That we have to deal with two limitations resides also from the procedure of forcible and sequential elimination of one of the two limits, hence reaching the obvious result that the un-eliminated limit will always be valid. Plainly, if we assume that we were now provided a wonder technology, we would grow aware that we still do not have an answer to the question of what to imitate with this technology. Likewise, if we suddenly understood what natural intelligence is, we would find that we do not have at the same time (simultaneously) a technological solution to imitate natural intelligence. Considering the current level of technological development, it is important to understand this dual limitation of AI in order to establish clear and separate research goals aimed at advancing from both directions to achieve the original goals assigned to artificial intelligence.

Furthermore, the motivational analysis of AI limitations was supported by our futuristic discussion structure on three levels. The first level approached via the existentialist dimension, i.e the perspective of knowing our limitations as a species on Earth, a discussion that ends with the certainty that humans have numerous limitations and that we know very little about what we would like to know, alongside a long series of unanswered questions. The second level was the launching of unanswered questions about the opportunity to design AIs that are superior to human beings. The third level focused on the usefulness of designing such AIs. Here we placed into discussion the idea of building *super-AI-slices*, i.e. those AIs that overcome natural intelligence in certain directions and that would be more useful at the societal level than a global artificial intelligence.

Motivated by the outstanding practical achievements of biocomputing, we focused on the technological challenge of decoding cellular communication through the understanding and development of the *natural language of the living cell* (N2LC), leading to the understanding and acquisition of novel information needed to monitor, coordinate, and direct cellular behavior, and we highlighted the topic of the so-called *invisible hand* outstretched by biocomputing towards AI.

Funding

This work was partial supported by a mobility grant of the Romanian Ministery of Research and Innovation, CNCS - UEFISCDI, project number PN-III-P1-1.1-MC-2018-2607, within PNCDI III, by the Faculty of Economy and Business Administration from "Alexandru Ioan Cuza" University Iasi, and by the University of Craiova.

Author contributions. Conflict of interest

The authors contributed equally to this work. The authors declare no conflict of interest.

Bibliography

[1] Abrahams, P.W. (1997). The World WithoutWork, In *P. Denning, R. Metcalfe, R. (Eds.), Beyond calculation: The nextfiftyyears of computing*, Santa Clara, CA: Springer-Verlag Telos, 135-147, 1997.

[2] Alles, M. (2018). Examining the role of the AIS research literature using the natural experiment of the 2018 JIS conference on cloud computing, *International Journal of Accounting Information Systems*, 31, 58-74, 2018.

[3] Alt, F.L. (1997). End-Running Human Intelligence, In *P. Denning, R. Metcalfe, R. (Eds.), Beyond calculation: The nextfiftyyears of computing*, Santa Clara, CA: Springer-Verlag Telos, 127 - 134, 1997.

[4] Badica, A.; Badica, C.; Ivanovic, M.; Buligiu I. (2016). Collective Profitability and Welfare in Selling-Buying Intermediation Processes. Computational Collective Intelligence, *Lecture Notes in Artificial Intelligence*, 9876, 14-24, Springer, 2016.

[5] Bell, G.; Gray J.N. (1997). The Revolution Yet to Happen, In *P. Denning, R. Metcalfe, (Eds.), Beyond calculation: The next fifty years of computing*, Santa Clara, CA: Springer-Verlag Telos, 127 - 134, 1997.

[6] Brown, J. N. A (2015). Once More, With Feeling: Using Haptics to Preserve Tactile Memories, *International Journal of Human-Computer Interaction*, 31(1), 65-71, 2015.

[7] Bruksos, R. (2005). *Turning Change into a Payday. Re-inventing Yourself Throughthe Eight Stage of Change.* Seatle: Training Consultants, 2005.

[8] Cearley, D.; Burke, B. (2018). *Top 10 Strategic Technology Trends for 2019*, Gartner Report, Available online: https://www.gartner.com/en/doc/3891569-top-10-strategic-technology-trends-for-2019, Accessed on 18 December 2018.

[9] ColdFusion TV. (2016, Mar. 29). *Nano-Biological Computing - Quantum Computer Alternative!* Video File, 2016,. Available online: https://youtu.be/xcHcNyC6O84, Accessed on 10 December 2018.

[10] Crawford, R. (2005). *Is God a Scientist? A Dialogue between Sinceand Religion*, Palgrave Macmillan, 2005.

[11] Data Science Institute (2017). *Researchers Store Computer Operating System and Short Movie on DNA*, Columbia University, 2017 Available online: https://datascience.columbia.edu/researchers-store-computer-operating-system-dna, Accessed on 10 December 2018

[12] Dekate, C.; Brethenoux, C.; Hare, J.; Chandrasekaran, A.; Govekar, M.; Ric, C. (2018). *Predicts 2019: Artificial Intelligence Core Technologies*, Gartner Report, 2018 Available online: https://www.gartner.com/doc/3894131?ref=mrktg-srch, Accessed on 18 December 2018.

[13] Denning, P.; Metcalfe, R. (Eds.); *Beyond calculation: The next fifty years of computing*, Santa Clara, CA: Springer-Verlag Telos, 1997.

[14] Descartes, R. (1864). *Discours de la Methode*, Emprimerie et Librairie Classiques, 1864.

[15] Dreyfus, H.L. (1972). *What computers can't do. A critique of Artificial Reason*, Harper & Row, 1972.

[16] Ecklund, E.H.; Scheitle, C.P. (2018). *Religion vs. Science: What Religious People Really Think*, Oxford University Press, 2018.

[17] Ecklund, E.H. (2012). *Science vs. Religion: what scientists really think*, Oxford University Press, 2012.

[18] Eden, A.H.; Steinhart, E.; Pearce, D.; Moor, J.H. (2012). Singularity Hypotheses: An Overview, In *A.H. Eden, J.H. Moor, J.H. Soraker, Steinhart, E. (Eds.), Singularity Hypotheses. A Scientific and Philosophical Assessment, The Frontiers Collection*, Springer-Verlag, 1-12, 2012.

[19] Erlich Y.; Zielinski, D. (2017). DNA Fountainenables a robust and efficient storage architecture, *Science*, 355(6328), 950-954, 2017.

[20] Eversmann, B.; Jenkner, M.; Hofmann, F.; Paulus, C.; et al.(2003). A 128 /spl times/ 128 CMOS Biosensor Array for Extracellular Recording of Neural Activity, *IEEE Journal of Solid-State Circuits*, 38(12), 2306-2317, 2003.

[21] Evolution News (March 10, 2017). *The Hottest New Computer Is: DNA, Evolution News*, Available online: https://evolutionnews.org/2017/03/hottest-new-computer-dna/, Accessed on 10 December 2018.

[22] Feigenbaum, E.A.; McCorduck, P. (1983). *The FifthGeneration: Artificial Intelligence and-Japan's Computer Challengetothe World*. Reading, Mass.: Addison-Wesley, 1983.

[23] Fiaidhi, J. (2011). Towards Developing Installable e-Learning Objects utilizing the Emerging Technologies in Calm Computing and Ubiquitous Learning, *International Journal of u- and e- Service, Scienceand Technology*, 4(1), 2011.

[24] Fotache, M.; Dumitriu, F.; Greavu-Serban, V. (2016). Differences in Information System Skills of Offshoring Source Markets and Destination Markets. A Romanian Perspective, *Transformations in Business & Economics*, Vol. 15(3C)/(39C), 42-59, 2016.

[25] Frankston, B. (1997); Beyond Limits, In *P. Denning, R. Metcalfe (Eds.), Beyond calculation: The nextfiftyyears of computing*, Santa Clara, CA: Springer-Verlag Telos, 43–58, 1997.

[26] Greenwood, P.E. (1967). Review of A View of Artificial Intelligence, by F.M. Tonge inProceedings A.C.M. National Meeting, 1966, *Computing Reviews*, 31, 1967.

[27] Hamidi, H.; Jahanshahifard, M. (2018). The Role of the Internet of Things in the Improvement and Expansion of Business, *Journal of Organizational and End User Computing*, 30(3), 24-44, 2018.

[28] Inside Human. (2018, Jul. 6); *Scientists Have Created Biological Computers Inside Living Cells*, Season 11. Video File. 2018, Available online: https://youtu.be/0hv3tzxb2IU, Accessed on 20 December 2018.

[29] Johnson, B. (2018). Beyond the Hard Drive: Encoding Data in DNA, *Future-Literacy*, Jul 10, 2018, Available online: https://medium.com/future-literacy/beyond-the-hard-drive-encoding-data-in-dna-1d5c2bad5289, Accessed on 18 December 2018.

[30] Kurzweil, R. (2011). *The Singularity is Near: When Humans Transcend Biology*, Old Saybrook: Tantor Media Inc, 2011.

[31] Lee, J. H.; Hwang, Y.; Cheon, K.A.; Jung H. I. (2012). Emotion-on-a-chip (EOC): Evolution of biochip technology to measure human emotionusing body fluids, *Medical Hypotheses*, 79 (2012), 827-832, 2012.

[32] Leonetti, M.; Dubois-Violette, E. (1998). Electrodynamic Instabilities in Biological Cells, *Physical Review Letters*, 81(9), 1977-1980, 1998.

[33] Loonam, J.; Eaves, S.; Kumar, V.; Parry, G. (2018). Towards digital transformation: Lessons learned from traditional organization. *Strategic Change*, 27, 101-109, 2018.

[34] Lungarella, M.; Iida, F.; Bongard, J.C.; Pfeifer R. (2007). AI in the 21st Century - With Historical Reflections, In *M. Lungarella, F. Iida, J.C. Bongard, R. Pfeifer R. (Eds), 50 Years of Artificial Intelligence. Essays Dedicated to the 50th Anniversary of Artificial Intelligence*, Springer, 1-8, 2007.

[35] Martin-Pena, M.L.; Diaz-Garrido, E.; Sanchez-Lopez J.M. (2018). The digitalization and servitization of manufacturing: A review on digital business model, *Strategic Change*, 27, 91-99, 2018.

[36] *Merriam-Webster's collegiate dictionary* (10th ed.) (1999). Springfield, MA: Merriam-Webster Incorporated, 1999.

[37] Molten, M. (2018). Biology Will Be theNext Great Computing Platform, *Wired Science*, 5.3.2018, Available online: https://www.wired.com/story/biology-will-be-the-next-great-computing-platform/, Accessed on 20 December 2018.

[38] Naisbitt, J. (1982). *Megatrends. Ten New Directions Transforming Our Lives*, WarnerBooks, 1982.

[39] Nicolau, D.V. Jr.; Lard, M.; Korten, T.; van Delft, F.C.M.J.M.; et al. (2016). Parallel computation with molecular-motor-propelledagents in nanofabricated networks, *Proceedings of the National Academy of Sciences*, 113 (10), 2591-2596, 2016.

[40] Nicolau, D.V.; Nicolau, D.V. Jr.; Solana, G.; Hanson, K.L.; et al. (2006). Molecular motors-based micro- andnano-biocomputationdevices, *Microelectronic Engineer*, 83(2006), 1582-1588, 2006.

[41] Norman, D. (1997). WhyIt's Good That Computers Don't Work Like the Brain, In *P. Denning, R. Metcalfe, R. (Eds.), Beyond calculation: The nextfiftyyears of computing*, Santa Clara, CA: Springer-Verlag Telos, 105-116, 1997.

[42] Perez, G. (2015). *Science vs. Religion: The Search for a Rational Approach*, Algora Publishing, 2015.

[43] Pop, P.; Araci, I.E.; Chakrabarty, K. (2015). Continuous-FlowBiochips: Technology, Physical Design MethodsandTesting, *IEEE Design & Test*, 32 (6), 8-19, 2015.

[44] Popescul, D.; Georgescu, M. (2015). Generation Y students in social: what do we know about them?, *BRAIN - Broad Research in Artificial Intelligence and Neuroscience*, 6(3-4), 74-81, 2015.

[45] Rosso, C. (2019); AI Created in DNA-Based Artificial Neural Networks. The intersection of artificial intelligence, syntheticbiology, andgenomics. *Psychology Today*, January 2019, Available online: https://www.psychologytoday.com/intl/blog/the-future-brain/201901/ai-created-in-dna-based-artificial-neural-networks, Accessed on 31 January 2019.

[46] Schwartz, J.T. (1986); The Limits of Artificial Intelligence. Technical Report #212, New York University, In *Encyclopedia of Artificial Intelligence*, 2 vols., John Wiley and Sons, 1986.

[47] Selfridge, O.G.; Neisser, U. (1960). Pattern recognition by machine. *Scientific American*, 203(2), 60-68, 1960.

[48] Starkermann R. (1993). The Functional Intricacy of Neural Networks A Mathematical Study, In *R.F. Albrecht, C.R. Reeves, N.C. Steele (eds), Artificial Neural Netsand Genetic Algorithms*, Springer, 1993.

[49] Tapscott, D. (1996). *The digital economy: Promise andperil in theage of networked intelligence*, McGraw Hill, 1996.

[50] Tofler, A. (1970). *Future Shock*, Bantam Books, 1970.

[51] Tonge, F.M. (1966). A View of Artificial Intelligence, *Proceedings A.C.M. National Meeting*, 379–382, 1966.

[52] Tugui, A. (2004). Reflections on the Limits of Artificial Intelligence. *Ubiquity*, ACM, 2004.

[53] Tugui, A.; Gheorghe, A. M. (2017). Review of the Book: The Singularity Is Near: When Humans Transcend Biology, by R. Kurzweil, *Transformations in Business & Economics*, 16(1), 252-256, 2017.

[54] Ulam, S. (1958). Tribute to John von Neumann, *Bulletin of the American mathematical society*, 64(2-3), 1-49, 1958.

[55] Vallee, R.(2009). An Unfinished Revolution? Heinz von Foerster and the Biological Computer Laboratory/BCL, 1958-1976, *Kybernetes*, 38 (1-2), 271-272, 2009.

[56] Vijayabanu, C.; Arunkumar, S. (2018). Strengthening the Team Performance through Personality and Emotional Intelligence: Smart PLS Approach. Scientific Annals of Economics and Business, 65(3), 303-316.

[57] Vinge, V. (1993); The coming technological singularity: how to survive in the post-human era, In NASA, *Lewis Research Center, Vision 21: Interdisciplinary Scienceand Engineering in the Era of Cyberspace*, Chapter 1, 1993.

[58] Von Foerster, H. (1966). *The Numbers of Man, Pastand Future. Biological Computer Laboratory Report 13.0*, Department of Electrical Engineering, University of Illinois, Urbana, IL 61801, 1966.

[59] Watanabe, C.; Naveed, K.; Touc, Y.; Neittaanmäki, P. (2018). Measuring GDP in the digital economy: Increasing dependence on uncaptured GDP, *Technological Forecasting and Social Change*, 137, 226-240, 2018.

[60] Winston, P. (1984); *Artificial Intelligence*, Adinson-Wesley Publishers, 1984.

[61] Zait, A. (2016). Conceptualization and Operationalisation of Specific Variables in Exploratory Researches - An Example for Business Negotion. *Scientific Annals of Economics and Business*, 63(1), 125-131, 2016.

[62] [Online]. *Love and Sex with Robot*, Robotics. Special Issue LSE 2016. Available online: https://www.mdpi.com/journal/robotics/special_issues/LSR, Accessed on 10 December 2018.

INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL
ISSN 1841-9836, e-ISSN 1841-9844, 14(2), 272-285, April 2019.

Ensemble Sentiment Analysis Method based on R-CNN and C-RNN with Fusion Gate

F. Yang, C. Du, L. Huang

Fushen Yang
Beijing University of Chemical Technology
Beijing 100029, China

Changshun Du*
Center for Information Technology
Beijing University of Chemical Technology
Beijing 100029, China
*Corresponding author: ducs@mail.buct.edu.cn

Lei Huang
School of Economics and Management
Beijing Jiaotong University
Beijing 100044, China

Abstract: Text sentiment analysis is one of the most important tasks in the field of public opinion monitoring, service evaluation and satisfaction analysis in the current network environment. At present, the sentiment analysis algorithms with good effects are all based on statistical learning methods. The performance of this method depends on the quality of feature extraction, while good feature engineering requires a high degree of expertise and is also time-consuming, laborious, and affords poor opportunities for mobility. Neural networks can reduce dependence on feature engineering. Recurrent neural networks can obtain context information but the order of words will lead to bias; the text analysis method based on convolutional neural network can obtain important features of text through pooling but it is difficult to obtain contextual information. Aiming at the above problems, this paper proposes a sentiment analysis method based on the combination of R-CNN and C-RNN based on a fusion gate. Firstly, RNN and CNN are combined in different ways to alleviate the shortcomings of the two, and the sub-analysis network R-CNN and C-RNN finally combine the two networks through the gating unit to form the final analysis model. We performed experiments on different data sets to verify the effectiveness of the method.
Keywords: Sentiment analysis, convolutional neural network, recurrent neural network, fusing gate.

1 Introduction

The basic task of sentiment analysis is to classify the polarity of a given text at the document, sentence or feature/aspect level and to determine whether the opinions expressed in the document, sentence or entity feature/aspect are positive, negative or neutral. The complexity of textual information leading to the detection of emotions in plain text makes for a challenging task. At present, the sentiment analysis algorithms with good results are all based on statistical learning methods. The performance of this method depends on the quality of feature extractions, while good feature engineering requires a high degree of expertise and is time-consuming, laborious, and offers poor mobility. The neural network approach can reduce the dependency of feature engineering.

Currently, some neural network-based methods have been used for sentiment classification tasks and Socher et al. [8–12] proposed modeling using the Recursive Neural Networks. These methods have proven to be effective in constructing sentence representations. However, recursive neural networks need to construct text as a data structure representation of a tree structure in order to capture the semantics of the sentence. Therefore, to a large extent, the rationality and validity of the text tree structure determines the performance of this type of method. At the same time, the construction of this text tree representation has a time complexity of at least O(n ^{2}), so when such a model is applied to a long sentence or document, significant time overhead is required. In addition, it is difficult to express the relationship between two sentences using a tree structure. Therefore, recursion is not suitable for long text modeling.

Another neural network model commonly used for natural language processing tasks, the Recurrent Neural Network (RNN), has a time complexity of O(n). The model uses the linguistic symbolic order input model in the text to model the text and store all of the text semantics in a fixed-size hidden state. Compared to other methods, RNN has a good ability to capture context information, which is useful for modeling long text and obtaining the semantics of long text. However, RNN is a biased model, and the closer to the end of the expanded word, the more information is retained than the previous word. It is well known that key features related to sentiment analysis may appear anywhere in the document, not just at the end of the text. Therefore, when the RNN is used to capture the semantic features of the entire input text, the change of the position of the key features will result in different degrees of effectiveness reduction, and may even completely ignore the important information, resulting in greatly degraded performance of the model.

At the same time, other work also uses the Convolutional Neural Network (CNN) for sentiment classification [6] . One reason why CNN was introduced into natural language processing tasks is that CNN can solve the problem of sequential deviation of features such as words. This is because it has translation invariance when using pooling operations such as maximum pooling, that is, the semantic features of different positions in the text are unbiased, regardless of the location of the feature. Therefore, compared with recursive or recurrent neural networks, CNN is more conducive to capturing semantic features that are independent of position and order in text. The time complexity of CNN is O(n). However, previous research work on CNN tends to use a simple convolution kernel such as a fixed window size [1,4] . When using such a kernel, it is difficult to determine the size of the window. Small windows can cause important information to be lost, while large windows lead to huge parameter spaces, making network training extremely difficult. At the same time, text analysis methods based on CNN can obtain the important features of text through pooling, but it is difficult to obtain the context information. The piecewise pooling strategy utilized by the sentiment analysis model in Du et al's work can partially alleviate the shortage of CNN [3] . However, modeling for long-distance dependence is still poor. Mathieu Cliche [2] uses CNN and long/short-term memory networks to model sentences separately. Although this method can improve the experimental results, it still cannot overcome the defects of CNN and RNN.

In order to solve the limitations of the above-mentioned recurrent neural network, recurrent neural networks and convolutional neural network models, some work has been attempted to integrate the Recurrent Neural Network and the convolutional neural network. For example, these works [5, 13] use convolutional cyclic neural networks to extract sequence features. Unlike other works, this paper proposes a sentiment analysis based fusion Recurrent-Convolutional Neural Network (R-CNN) [7] and Convolutional-Recurrent Neural Network (C-RNN) sentiment analysis method. Firstly, RNN and CNN are combined in different ways to alleviate the shortcomings of the two. The sub-analysis networks R-CNN and C-RNN are constructed respectively. Lastly, the final analysis and analysis model is composed by combining the two networks with a fusion

gate. We performed experiments on different data sets to verify the effectiveness of the method.

2 Sentiment analysis method based on R-CNN and C-RNN with fusion gate

2.1 R-CNN-based text sentiment feature extraction model

Firstly, this paper proposes a deep neural model based on the R-CNN to capture the semantics of the text, and uses the obtained semantic features as the characteristics of the sentiment analyzer to analyze the sentiment orientation. Figure 1 shows the network structure of the model in this section. The input to the network is the text S, which consists of the word sequence $w_1, w_2, \cdots, w_n$ of the text.

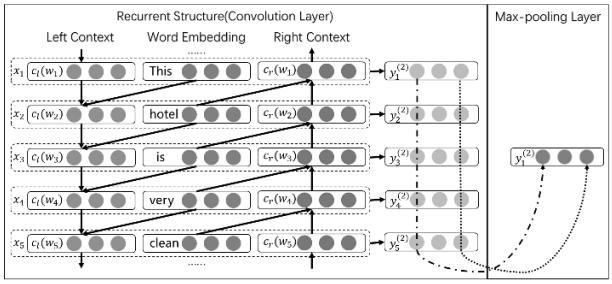


Figure 1: structure of recurrent-convolution neural network

Semantic feature extraction of words

In the text, the meaning of each linguistic symbol is related to the context in which it is located. The contextual location of the word can help the model to obtain a more precise meaning of the word in a particular scene. In order to enable the semantic features extracted by the neural network to fuse the context of the word, this paper uses the R-CNN model to extract features from the customer's comment text. The model first uses a Bidirectional Recurrent Neural Network to capture the contextual information of the word and combine it with the word embedding representation of the word itself to obtain an enhanced representation of the word. The definitions $c_l(w_i)$ and $c_r(w_i)$ represent the left context information (the loop is expanded from the front to the back) and the right context information (expanded from the back), respectively, of the word w_i. $c_l(w_i)$ and $c_r(w_i)$ are realvalued dense vectors of dimension c, and the calculation processes are expressed as Equation 1 and Equation 2, respectively.

$$c_l(w_i) = f(W^l c_l(w_{i-1}) + W^{sl} e(w_{i-1})) \tag{1}$$

$$c_r(w_i) = f(W^r c_r(w_{i+1}) + W^{sr} e(w_{i+1})) \tag{2}$$

$e(w_i)$ represents the word embedding representation of the i-th word w_i in the input text sequence. $c_l(w_i - 1)$ represents the left context information of the previous word w_{i-1}. W^l is a parameter matrix that converts the forward-expanded hidden layer state (left context information feature) of the bidirectional cyclic neural network into the next hidden layer state. W^{sl} is a parameter matrix that embeds the information of the current word into the left context

information feature added to the next word. f is a nonlinear activation function. The right context information feature $c_r(w_i)$ is calculated in a similar manner, as shown in Equation 2. In order to conveniently calculate the context information of the first and last words of the input text, a rightmost context feature $c_r(w_n)$ of all text sharing is set to estimate the leftmost context information feature $c_l(w_1)$.

After computing the text sequence using Equations 1 and 2, the left context information feature of the current word captures all semantic information that is expanded from front to back to the word. The right context information feature captures all the semantics that are expanded from back to front to the word. For example, in Figure 1, $c_l(w_5)$ retains the semantics of the context "...this hotel bathroom comparison" made up of all the previous words on the left side of the word “clean." $c_r(w_1)$ retains the semantics of the phrase “the hotel bathroom is clean..." on the right and end of the word “this home". After obtaining the contextual semantic features of the current word, the contextual semantic feature is merged with the word embedded representation of the word itself to obtain the final representation of the current word, and the formal representation is as follows:

$$x_i = [c_l(w_i); e(w_i); c_r(w_i)] \tag{3}$$

As shown in Equation 3, the final representation x_i of the word w_i consists of a splicing of the left context vector $c_l(w_i)$, the word embedding $e(w_i)$, and the right context vector $c_r(w_i)$. In this way, the word representations learned by the model contain rich word context information. Compared to traditional language models or neural network models that use only fixed window sizes, these models use only part of the textual information around the word when processing each language symbol. The RCNN model can model the entire text sequence and preserve the structural information such as the order of the words. The context information can better eliminate the ambiguity of the meaning of the word w_i. At the same time, the RNN structure forward scan obtains all left context semantic features c_l, and the backward scan obtains all right context features c_r, with a total time complexity of O(n), which has high efficiency.

After obtaining the representation x_i of the word w_i the linear activation function tanh transforms the x_i and outputs it to the next layer, namely:

$$y_i^{(2)} = tanh(W^{(2)}x_i + b^{(2)}) \tag{4}$$

$1 \leq i \leq |S|$. $y_i^{(2)}$ is a potential semantic vector that integrates word and context information, so each dimension can be thought of as an element that contains a certain type of semantic information. These semantic elements are useful for extracting features related to sentimental tendencies throughout the input text.

Obtaining the Semantic Features of the Full Text from the Semantic Features of Words

Directly using the representation of a single word in the text to analyze the sentiment orientation of the entire comment is still very difficult, so it is also necessary to use the representation of the words to calculate the representation of the entire comment text. From the perspective of CNN, the aforementioned RNN structure can be considered as the convolutional layer of CNN. When the representation of all words is calculated, the maximum value pooling operation is used to obtain the semantic features of the full text from all word representations of the comment text sequence, as shown in Equation 5:

$$y^{(3)} = \max_{i=1}^{|S|} y_i^{(2)} \tag{5}$$

The maximum value among the features obtained by extracting each convolution kernel convolution operation is different from the maximum pooling operation employed by other work. In the R-CNN model, the maximum pooling operation extracts the maximum value for all word representations of the entire comment text. That is, the maximum value of the k-th dimension of all the word semantic features $y_i^{(2)}$ is selected as the value of the full-text semantic feature $y^{(3)}$ of the k-th semantic element. After the maximum pooling operation, the comment texts with different lengths are converted into fixed-length full-text semantic vectors, which can better preserve the important semantic information of the entire text. There are many types of pooling operations in convolutional neural networks, such as average pooling, minimum pooling, and the like. As mentioned above, in the whole comment, some emotion-related words and their combined semantics are the most relevant features of the sentiment analysis task, and are very effective features for judging the emotional tendency of comments. Extracting these important features can improve the performance of the sentiment analysis model. Therefore, in the model of this section, the maximum pooling operation is still used. The pooling layer uses the output of the RNN as input, and the time complexity of the pooling layer is also O(n). Therefore, the time complexity of the cascaded RCNN sentiment feature extraction model is still O(n), which maintains high efficiency.

2.2 Text sentiment feature extraction model based on C-RNN

In the previous section, this paper proposes the use of RCNN for feature sentiment feature extraction of comment texts. In this section, this article will examine another combination of convolution and recurrent neural networks. That is to say, the convolution operation is performed first, and then the combination of the loop structure is used to propose a feature extraction model based on C-RNN.

Convolution operation segmentation extraction of local emotional features of review text

A standard convolutional neural network usually consists of a convolutional layer and a pooled layer. As mentioned above, the standard convolutional neural network can extract the combined features of words and has translation invariance. These combined features can be extracted anywhere in the text, and convolution operations can be extracted. In the model presented in this section, the input is a matrix of words embedded in the text, and each line of the matrix is a vectorized representation of a linguistic symbol. The convolution kernel slides in the direction in which the text is expanded, that is, the width of the input matrix (the dimension of the word vector) coincides with the width of the convolution kernel. Assuming that the height of the convolution kernel is w, and the width and word vector dimensions are both d, the convolution kernel can be represented as a matrix $W \in R^{w \times d}$. Let the vectorization of the i-th language symbol in the input be represented as s_i, and the input text can be represented by the matrix $S = (s_1^T, s_2^T, \cdots, s_{|S|}^T)$. Then the convolution operation can be expressed as follows:

$$c^j = W \bigotimes S_{j:j+w-1} \tag{6}$$

$1 \leq j \leq |S| - w + 1$, c^j is the eigenvalue extracted by the convolution operation between the convolution kernel starting from the word j and the convolution kernel height being the window.

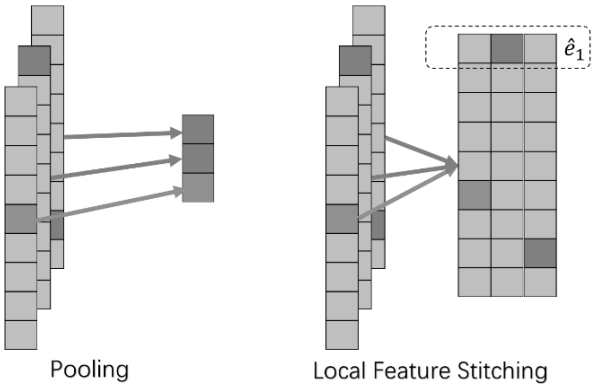


Figure 2: Local feature stitch and pooling

The feature extraction of text by relying solely on a convolution kernel is not comprehensive. In order to extract more abundant information from the text, multiple different convolution kernels are usually used. These convolution kernels can be expressed as a three-dimensional tensor $\hat{W} = \{W_1, W_2, \cdots, W_n\}$, and the convolution operation of the convolutional layer can be expressed as follows:

$$c_i^j = W_i \bigotimes S_{j:j+w-1} \tag{7}$$

$1 \leq i \leq n$, The input text is subjected to the ith convolution kernel convolution operation to obtain the feature vector $c_i = \{c_i^1, c_i^2, \cdots, c_i^{|S|-w+1}\}$. Then all convolution kernels can get a total of n feature vectors. In the model proposed in this section, all convolution kernels are of the same size, so the same comment text after convolution operation will get n vectors with the same dimension and containing the local semantic features of the comment text. The same dimension of these feature vectors can be regarded as different types of features of the same part in the comment, that is, each local semantic feature $\hat{e}_i$ is composed of n features. Different local semantic features may have different n feature strengths. Unlike the traditional convolutional neural network model, the convolutional neural network model in this section no longer uses the pooling layer, but splices the feature vector after the convolution operation into the sequence feature output to the next layer, as shown in Figure 2.

RNN fusion text sentence structure features

After obtaining the local features of the comment text by using the convolutional layer, the local feature is regarded as a text sequence input to the cyclic neural network, and the long-distance dependence of the comment text can be modeled, and the structural features of the sentence are incorporated into the local features of the text.

In this paper, the bidirectional RNN based on LSTM computing nodes is used to perform text representation learning on the stitched feature sequences, and then the feature vectors learned in the two directions are stitched together as a vector representation of the text. Thus, the semantics represented by the feature vector are more comprehensive and rich than RNN. The t-th local feature obtained after splicing is expressed as $\hat{e}_t$, and the one-way calculation process of incorporating the structural features of the text sentence through RNN can be expressed as follows:

$$\begin{cases} i = \sigma(U^i \hat{e}_t + W^i h_{t-1} + b_i) \\ f = \sigma(U^f \hat{e}_t + W^f h_{t-1} + b_f) \\ o = (U^o \hat{e}_t + W^o h_{t-1} + b_o) \\ \tilde{C} = tanh(U^c \hat{e}_t + W^c h_{t-1} + b_c) \\ C_t = C_{t-1} \bigotimes f + \tilde{C} \bigotimes i \\ h_t = o \bullet tanh C_t \end{cases} \tag{8}$$

h_t is the hidden state of the computing node. $\tilde{C}$ indicates that the candidate cell transition is calculated from the previous time hidden state h_t and the current input x_t. C_t indicates the cell state, which is obtained by the forgetting gate and input gate weighting calculation by the cell state C_{t-1} of the $t-1$ step and the cell-selective $\tilde{C}$. h_t is the current output of the computing node, and is also the current hidden state. It is the amount of information that the cell state C_t is finally output through the output gate selection. The local feature of the review text S is forwardly expanded through the RNN to obtain the hidden layer state $\overrightarrow{h_S}$ of the forward sentence structure feature. The backward expansion results in the hidden layer state $\overleftarrow{h_S}$ of the backward sentence structure feature. What is obtained after splicing is the final feature vector h_S representing the input comment text, that is, $h_S = [\overrightarrow{h_S}, \overleftarrow{h_S}]$.

Therefore, the overall structure of the C-RNN-based text sentiment feature extraction model proposed in this paper can be represented as shown in Figure 3. The computational time complexity of the convolutional layer and the cyclic layer is O(n), so the total time complexity of the model proposed in this section is still maintained at O(n), which maintains the efficiency of the model.

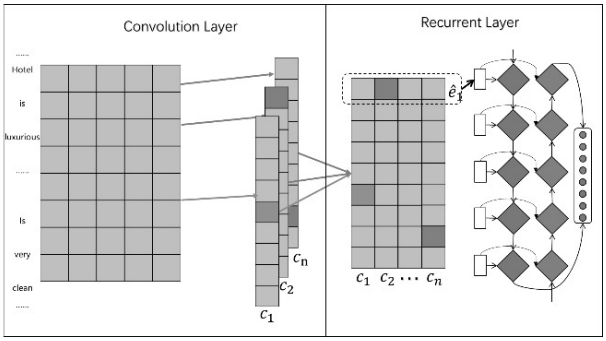


Figure 3: Text sentiment feature extraction model based on convolution-recurrent neural network

2.3 Sentiment analysis model based on R-CNN and C-RNN of with fusion gate

Feature fusion based on gating unit

After R-CNN and C-RNN respectively extract the emotional features of the review text, the features extracted by the two networks need to be integrated and input into the sentiment orientation analyzer to obtain the final sentiment analysis result.

For comments in different segments, there is a difference in the characteristics associated with emotional orientation, that is, emotional tendencies have domain dependence. The comment texts in the same segment have different styles, so there are some differences in the sentiment orientation characteristics. The two combined models of RNN and CNN proposed in this paper

are rich in features. However, in different fields, the same feature has different effects on the discrimination of sentiment orientation. Too many features will introduce noise into the sentiment orientation, which degrades the performance of the analyzer. However, relying on manual screening of these features is very difficult. Therefore, an automatic method is needed to select these features to make domain, text adaptability, remove redundant information, and ensure the effectiveness of the sentiment orientation analyzer.

This paper proposes to use the fusion gate to automatically filter and fuse the two features extracted by R-CNN and C-RNN. The process of fusion can be represented in Figure 4.

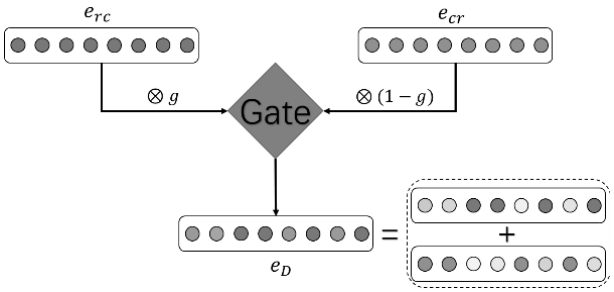


Figure 4: R-CNN combines C-RNN extract features based on gate unit

The fusion gating unit calculates, based on the two characteristics currently input, how many features can be passed through the fusion gating unit, that is, how many original features need to be retained for the two features of the output features. For convenience of representation, the features extracted by the R-CNN described above are denoted as e_{rc}, and the features extracted by the C-RNN are denoted as e_{cr} . The calculation process of the control ratio g of the fusion gating unit to the feature passing is as follows:

$$g = \sigma(U^g e_{rc} + W^g e_{cr} + b^g) \tag{9}$$

U^g and W^g are the weight parameter matrix of the fusion gating unit connected to e_{rc} and e_{cr}, respectively, and b^g represents the offset. In order to simplify the calculation, the fusion gating unit uses the same control ratio for feature extraction of the two extracted features, and the feature fusion calculation process can be expressed as follows:

$$e_D = g \bigotimes e_{rc} + (1-g) \bigotimes e_{cr} \tag{10}$$

e_D is a feature vector that characterizes the input commentary sentiment feature obtained by fusing the two features of the fusion gating unit.

Softmax classifier gets comments emotional sentiment

After obtaining the emotional feature e_D of the review text, the softmax classifier is used to classify the emotional sentiment of the review based on the feature vector, and obtain the emotional tendency of the review.

Set the network parameter W_e of the softmax classification layer, and the bias term is b_e, then the neural network output can be expressed as:

$$o = f(W_e e_D + b_e) \tag{11}$$

f represents the activation function. Then the probability that the entered text sentiment tends to i is:

$$p(i|\theta) = \frac{e^{o_i}}{\sum_{j=1}^{N} e^{o_j}} \tag{12}$$

θ represents all parameters of the neural network, o_i represents the value of the i-th item of the output vector, and N represents the number of categories of the text. Let the sample set be expressed as Ω, then the model's optimization objective function can be calculated by the following formula.

$$L_{sen} = \sum_{i=1}^{|\Omega|} -logp(y_i|S_i, \theta) + \lambda \parallel \theta \parallel_2^2 \tag{13}$$

λ is the parameter of the regular term. In the actual experiment, we use the stochastic gradient descent method to optimize the objective function, then the parameter θ is updated in the way:

$$\theta = \theta - \alpha \frac{\partial L}{\partial \theta} \tag{14}$$

α is the learning rate.

3 Experiment settings

3.1 Data set

Three data sets were used primarily in the experiment. The first is the Chinese Hotel Data Collection (Ctrip Hotel), with a corpus size of 10,000, including 7,000 positive evaluation samples and 3,000 negative evaluation samples. The second dataset is the English dataset, a film review dataset released by Pang and Lee in 2005. The dataset contains a total of 10,662 commentaries on the film, with emotional sentiment being half positive and half negative emotionally biased samples. The sentiment orientation tab of the review reflects the overall emotional sentiment of the reviewer's comments. This paper divides the data set into a training set, a validation set, and a test set, which contain 8530, 1066, and 1066 samples, respectively, with half having positive and half having negative emotional tendencies. The emotional polarity of each sentence is in the range of [0,1]. The smaller the score, the more the emotion tends to be negative. Otherwise, the emotion tends to be positive. The emotional scores of all sentences in the data set are manually labeled and then averaged. It has good reliability. In order to make the experiment closer to the real production environment, and to verify the effectiveness of the sentiment analysis method based on the R-CNN and C-RNN proposed in this paper, the third data set is used in this paper. It is the review text (Dianping for short) in different fields that we crawled from the public service website (http://www.dianping.com/), including data on six segments of food, hotel, movie, entertainment, marriage, and home improvement. And based on the scoring information in the comments, the comments are divided into different emotional tendencies. This paper selects 30,000 reviews as the training set and 10,000 as the test set.

3.2 Data pre-processing

For the Chinese data set, the Chinese word segmentation package NLPIR developed by the Chinese Academy of Sciences is first used for Chinese word segmentation. The English data itself is an independent word, so there is no need for word segmentation. Since the minibatch training model is used during training (multiple samples are learned at a time, the text length of multiple samples may not be the same). At the same time, the use of C-RNN to extract features of different texts needs to ensure the uniformity of dimensions. After convolution, the splicing can guarantee the feature dimension and therefore needs to complete the operation for the length of the text. Since the length of the text of the comment is inconsistent, the longest sentence length l_max is calculated first. For sentences with a sentence length less than l_max, the text must be unified with the $< \backslash s >$ symbol to the length l_max (the vector of $< \backslash s >$ is always set to 0), thus unifying the text length. The length of the unified text can improve the calculation efficiency, and when the length of the data is uniform, the calculation time overhead can be effectively reduced. At the same time, in order to ensure the feature extracted at the beginning and the end of the text during the convolution process, a certain number of $< \backslash s >$ corresponding to the convolution kernel is added at the beginning and end of the longest text as Padding.

3.3 Pre-training of word embedding

Word embedding is required before formal training of the model. Word embedding acts as a distributed representation of words as an input suitable for neural networks. Many current studies have shown that executing word embedding pre-training on a large-scale corpus, and then applying the obtained word embedding to subsequent training, can speed up the convergence of neural network models and achieve a better local optimal solution. In this paper, the word2vec algorithm is used to pre-train word embedding. The word embedding of this algorithm shows better performance in many natural language processing tasks, and it has higher efficiency. This paper chooses the Skip-gram model and the Negative Sampling model to pre-train the word embedding of Chinese and English words. The pre-training of Chinese word embedding uses the text content crawled on Baidu Encyclopedia, and the pre-training of English word vectors is performed on the New York Times corpus.

3.4 Setting of experimental parameters

In the training optimization process of the model, the Adam optimizer is used to train and optimize the parameters of the model. The parameters of the Adam optimizer are set by the author. In this paper, the model mainly has the following hyperparameters: the dimension $\boldsymbol{d}$ of the word vector, the number $\boldsymbol{n}$ of convolution kernels in the C-RNN, the dimension $\boldsymbol{N_{rc}}$ of the hidden state in the loop structure in the R-CNN, and the dimension $\boldsymbol{N_{cr}}$ of the hidden state in the C-RNN. In order to obtain the optimal hyperparameter setting, this paper uses grid-search to determine the value of some hyperparameters. The dimension $\boldsymbol{N_{rc}}$ of the hidden state in the loop structure in the R-CNN and the dimension $\boldsymbol{N_{cr}}$ of the hidden state in the C-RNN are selected from { 50, 100, 200, 300 }. The number n of convolution kernels takes values in { 100, 150, 200 }. In the experiments in this paper, multiple experiments were performed using these parameters, and then the average of the results was obtained.

4 Results and analysis

In order to verify the effectiveness of the proposed R-CNN and C-RNN integrated sentiment analysis method based on the fusion gating unit, the method of this paper is compared with some mainstream sentiment analysis baseline methods.

4.1 Comparison method

In order to verify the validity and correctness of the proposed model, this paper selects a model based on traditional methods and a neural network method such as RNTN proposed by Richard Socher et al. as a baseline method. The first method of comparison is the naive Bayesian method (abbreviated as NB) using the word bag feature. The second method is to use the word bag feature as input to perform emotion classification using a support vector machine (abbreviated as SVM) classifier. The third method is the naive Bayesian method by using the bag feature obtained from the binary grammar language model (abbreviated as BiNB). The fourth method is to use the average word vector of the sentence as the input feature and use the fully connected network as the classifier (abbreviated as VecAvg). The fifth method is the Recurrent Neural Network (RNN). The sixth method is a recurrent neural network (MV-RNN) with a semantic transformation matrix [1] . The seventh method is based on a tensor-based cyclic neural network (RNTN). The last comparison method is the traditional convolutional neural network.

4.2 Analysis of experimental results

It can be seen from the results in Table 1 that the neural network method generally has higher performance than the conventional method. Especially in the five-level sentiment analysis with finer granularity, the neural network method can obtain the key features of the text well. An important reason for the BiNB method to achieve better results is that the binary grammar model considers a certain combination of semantics, but with it comes significant computational overhead. At the same time, compared with the cyclic neural network method, the use of maximum pooling in convolutional neural networks can automatically extract the features most relevant to sentiment analysis tasks, and has positive significance for text sentiment analysis tasks, so it has achieved good results. Traditional methods such as BiNB can only extract combined features from adjacent words, and traditional convolutional neural networks cannot model grammatical structures. The segmented convolutional neural network method has the best effect on both datasets because it simulates the grammatical structure information of the text, supplements the original emotional words, and extracts the combined semantic features of different positions.

In the Dianping data, the data comes from different fields. We compare the performance of different methods in this data to fully demonstrate the effectiveness of the proposed R-CNN and C-RNN integrated sentiment analysis method based on the fusion gating unit. It can be seen from Table 2 that the method proposed in this paper achieves the best results. The comments crawled from the Dianping contain different subdivisions. The comment text in each field is relatively small, and the comment text itself is relatively short. Therefore, samples in different fields have a certain degree of sample sparseness. This need model can effectively extract emotion-related features from the text in order to correctly judge the emotional tendency of the text. This paper also reduces the number of samples in each field, making the data sparse more serious. It can be seen that all methods have performance degradation problems to varying degrees, but the method proposed in this paper still achieves good results. The stability of the proposed method in the case of multi-domain data sparseness is fully explained.

Table 1: Sentiment analysis precision on different data sets(%)

Model	**Stanford**		**Ctrip**
	Two levels sentiment	**Five levels sentiment**	
NB	81.8	41.0	80.2
SVM	79.4	40.7	86.7
BiNB	83.1	41.9	85.9
VecAvg	80.1	32.7	82.1
RNN	82.4	43.2	87.8
MV-RNN	82.9	44.4	87.6
RNTN	**85.4**	45.7	89.3
CNN	81.9	45.6	88.5
R-CNN&C-RNN	**85.8**	**46.1**	**89.8**

Table 2: Sentiment analysis precision on Dianping data sets(%)

Model	**Two levels sentiment**	**Five levels sentiment**	**Two levels sentiment(reduced)**
NB	72.1	35.7	66.1
SVM	72.6	33.2	61.2
BiNB	75.2	36.6	62.4
VecAvg	74.1	36.3	62.3
RNN	77.3	40.2	69.7
MV-RNN	78.0	42.2	73.1
RNTN	80.3	42.6	75.7
CNN	79.1	41.5	73.5
R-CNN&C-RNN	**83.5**	**47.6**	**80.6**

Table 3: Sentiment analysis precision after adding negative words on the dataset(%)

Model	**Negated Positive**	**Negated Negative**
BiNB	39.0	27.6
VecAvg	16.5	17.9
RNN	43.3	44.2
MV-RNN	62.4	56.2
RNTN	81.4	72.6
R-CNN&C-RNN	**81.7**	**77.2**

In order to further illustrate the effectiveness of the integrated analysis method of R-CNN and C-RNN based on the fusion gate, this paper selects some samples from the Dianping data set. Adding negative words to the sample converts the emotional tendency of the sample into opposing emotions, called negative positive and negative negative. For example, the “food in this restaurant is delicious" is changed from positive to negative as “the food in this restaurant is not delicious". Turning “this film is ugly" from negative to positive by adding a negative to the original negative (negative negative), “this movie is not ugly." It can be seen that such a change is very subtle for the change of the review text. When the negative word is added to the positive sample, the emotional tendency changed to negative absolutely. However, when the negative word is added to the negative sample, the change of the emotional tendency is tiny. These samples can test the effectiveness and stability of sentiment analysis methods for emotion extraction, and test whether the model can capture emotional details in the text.

The results of the test are shown in Table 3. We can see that the proposed method is superior to all baseline methods. The experimental results show that our model can better capture the emotional features of the review text from the details.

5 Conclusion

In this paper, the performance of traditional text sentiment analysis methods depends on the quality of feature extraction, while good feature engineering requires a high degree of expertise, and is time-consuming, laborious, and affords poor mobility. The neural network approach can reduce the dependency on feature engineering. RNN can obtain context information, but the order in which the features appear in the text will cause the model modeling to be biased. The text sentiment analysis method based on CNN can obtain important features of text through pooling, but it is difficult to obtain context information, and the problem of poor modeling ability for long distance dependence is poor. In this paper, a sentiment analysis method based on fusion gating unit R-CNN and C-RNN is proposed. Firstly, CNN and RNN are combined in different ways to alleviate the shortcomings of the two, and then the sub-analysis networks R-CNN and C-RNN are constructed respectively. Lastly, the final analysis model is composed by combining two networks through the fusion gating unit.

The model proposed in this paper has carried out more detailed experiments on different data, which proves that the proposed method has strong adaptability in different fields. It is more able to extract emotion-related features from text, and maintains good validity in shorter texts and fewer samples. At the same time, the experimental results of adding negative words in the comment text show that the R-CNN and C-RNN integrated sentiment analysis methods based on the fusion gating unit can better capture the emotional features of the review text from the details and that it has good stability.

Author contributions. Conflict of interest

The authors contributed equally to this work. The authors declare no conflict of interest.

Bibliography

[1] Collobert, R.; Weston, J.; Bottou, L.; et al. (2011). Natural language processing (almost) from scratch, *Journal of Machine Learning Research*, 12, 2493-2537, 2011.

[2] Cliche, M. (2017). BB_twtr at SemEval-2017 Task 4: Twitter Sentiment Analysis with CNNs and LSTMs, *arXiv preprint arXiv*, 1704.06125, 2017.

[3] Du, C.; Huang, L. (2019). Sentiment Analysis Method Based On Piecewise Convolutional Neural Network and Generative Adversarial Network, *International Journal of Computers Communications & Control*, 14(1), 7-20, 2019.

[4] Kalchbrenner, N.; Blunsom, P. (2013). Recurrent convolutional neural networks for discourse compositionality, *arXiv preprint arXiv*, 1306.3584, 2013.

[5] Kalchbrenner, N.; Grefenstette, E.; Blunsom, P. (2014). A convolutional neural network for modelling sentences, *arXiv preprint arXiv*, 1404.2188, 2014.

[6] Kim, Y. (2014). Convolutional neural networks for sentence classification, *arXiv preprint arXiv*, 1408.5882, 2014.

[7] Lai, S.; Xu, L.; Liu, K.; et al. (2015). Recurrent Convolutional Neural Networks for Text Classification, *AAAI*, 333, 2267-2273, 2015.

[8] Luong, T.; Socher, R.; Manning, C. (2013). Better word representations with recursive neural networks for morphology, *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*, 104-113, 2013.

[9] Socher, R.(2014). Recursive deep learning for natural language processing and computer vision, *Stanford University*, 2014.

[10] Socher, R.; Chen, D.; Manning, C.D.; et al.(2013). Reasoning with neural tensor networks for knowledge base completion, *Advances in neural information processing systems*, 926-934, 2013.

[11] Socher, R.; Huval, B.; Manning, C.D.; et al. (2012). Semantic compositionality through recursive matrix-vector spaces, *Proceedings of the 2012 joint conference on empirical methods in natural language processing and computational natural language learning. Association for Computational Linguistics*, 1201-1211, 2012.

[12] Socher, R.; Perelygin, A.; Wu, J.; et al. (2013). Recursive deep models for semantic compositionality over a sentiment treebank, *Proceedings of the 2013 conference on empirical methods in natural language processing*, 1631-1642, 2013.

[13] Shi, B.; Bai, X.; Yao, C. (2017). An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition, *IEEE transactions on pattern analysis and machine intelligence*, 39(11), 2298-2304, 2017.

Author index