ANNALES MATHEMATICAE ET INFORMATICAE

VOLUME 56. (2022)

INSTITUTE OF MATHEMATICS AND INFORMATICS
ESZTERHÁZY KÁROLY CATHOLIC UNIVERSITY
HUNGARY, EGER

Selected papers of the 2nd Conference on Information Technology and Data Science

The conference was organized by
Faculty of Informatics, University of Debrecen, Hungary,
May 16–18, 2022

Conference General Chair
András Hajdu

Program Committee Chair
István Fazekas

HU ISSN 1787-6117 (Online)

A kiadásért felelős az
Eszterházy Károly Katolikus Egyetem rektora
Megjelent a Líceum Kiadó gondozásában
Kiadóvezető: Dr. Nagy Andor
Műszaki szerkesztő: Dr. Tómács Tibor
Megjelent: 2022. december

Contents

Annales Mathematicae et Informaticae
56 (2022) pp. 1–9
DOI: https://doi.org/10.33039/ami.2022.12.006
URL: https://ami.uni-eszterhazy.hu

Benchmarking Redis and HBase NoSQL Databases using Yahoo Cloud Service Benchmarking tool

Mustafa Alzaidi, Aniko Vagner

University of Debrecen-Faculty of Informatics
mustafa.alzaidi@inf.unideb.hu
vagner.aniko@inf.unideb.hu

Abstract. The Not Structured Query Language (NoSQL) databases have become more relevant to applications developers as the need for scalable and flexible data storage for online applications has increased. Each NoSQL database system provides features that fit particular types of applications. Thus, the developer must carefully select according to the application's needs. Redis is a key-value NoSQL database that provides fast data access. On the other hand, the Apache HBase database is a column-oriented database that offers scalability and fast data access, is a promising alternative to Redis in some types of applications. In this research paper, the goal is to use the Yahoo Cloud Serving Benchmark (YCSB) to compare the performance of two databases (Redis and HBase). The YCSB platform has been developed to determine the throughput of both databases against different workloads. This paper evaluates these NoSQL databases with six workloads and varying threads.

Keywords: Redis, HBase, YCSB, Benchmarking, NoSQL Databese

1. Introduction

A growing number of NoSQL databases are being developed and used. The promise of quicker and more efficient throughput compared to older Relational Database Management Systems (RDBMS) is one of its most compelling features[14]. There are several advantages to using NoSQL databases for cloud computing, including the ability to rapidly scale vertically and horizontally as needed and the easiness of application development[8]. However, big data and online application developers

Submitted: November 27, 2022
Accepted: December 3, 2022
Published online: December 28, 2022

should be aware that NoSQL databases are not usually equal when it comes to performance [6]. Because NoSQL systems are not yet mature and evolving at various paces, database managers must pick carefully between NoSQL and relational databases based on their demands regarding consistency, security and scalability, performance, prices, and other factors[15]. Choosing a NoSQL system might be a challenge for web application developers because of the large variety of open-source and freely accessible NoSQL systems. In other words, a peer-to-peer comparison of NoSQL systems according to the application activity scenarios to identify the most significant match for different situations would be an appropriate next step. A benchmark in this context refers to a performance assessment of NoSQL solutions that have been suggested or have been deployed. Then, compare the performance of different NoSQL databases; it is necessary to utilize experimental interactions that simulate comparable behavior or activities, as could be the case with applications behavior. Selecting a NoSQL system in this manner can be more appropriate for certain types of user interaction and provide better performance and efficiency than a competitor's systems. key-value, wide column, graph, and document databases are all examples of NoSQL databases[12, 15]. Key-value stores are collections of registers identifiable by a unique key [3]. Usually, this type of NoSQL system is used as a layer that provides cash for the data with time-consuming access[4]. Some researchers[2] use the key-value store when the application needs to retrieve the stored object based on one field value. Javascript and Binary Object-Notations (JSON and BSON) is a kind of document-oriented data[13]. Document-based databases provide more flexibility in terms of schema compared to RDBMS. They store the data in objects format in a similar manner to how programming language logically treats objects. The schema-less model enables the developer to store different types of objects in the same storage entity. This flexibility gave the ability to rapid application development [7]. Document store databases can work well on distributed systems that provide cheaper horizontal scaling as the application needs. Databases like MongoDB, CouchDB, and others fall within this category. The success of Google with BigTable seems to have sparked the development of column stores [5]. The column store databases stores the tables records fields separately, such that subsequent values of that property are saved sequentially [1]. Wide-Column database systems are built on a hybrid method that makes use of both the descriptive qualities of relational databases and the structure of different key-value stores [15]. Accumulo, Cassandra, as well as HBase are fall in this category. Graph databases may be used to store objects data, as well as all connections between them [15]. In this way, Graph databases make use of nodes and edges, the two notions from Graph theory. For example, a foreign or primary keys link between two nodes is an edge in the data domain. Neo4J and OrientDB are two good examples[11]. In this paper, we did use Yahoo Cloud Service Benchmarking (YCSB) tool benchmark Redis and HBase databases. We did the test with six different workload scenarios for each workload, and we recorded ten results by adding a new thread each time with ten threads till the last text.

2. Redis NoSQL database

Redis is an open-source in-memory key-value store database that is very customizable and claims to be extremely quick in terms of performance. VMware initially maintained it; later, Pivotal Software has taken over as the company that is sponsoring its development. Typically, the databases in Redis is specified by a numerical value. The number of databases is set at 16 by default, although this may be changed as a custom configuration. It is more customizable than a generic key-value structure in terms of data organization. For example, a value in Redis may be saved as a string, a list of strings with insertions at the beginning and end of the list. Furthermore, searching for objects towards the two ends of a huge list is incredibly quick, but querying for an item in the center of a large list is much more time-consuming. The collection of strings stored in Redis does not allow duplication, which implies that adding the same key (string) more than once will result in just one copy of the collection. The operations of adding and removing only need a constant amount of time (O(1)). Redis provides other structures like Hash, Set, and Sorted Set. Hash is referred to by a unique key and can store a set of unique fields, where each field can have one value. Hash provides high-speed data access in comparison to other structures. For instance, in comparison to List, even a colossal Hash can retrieve any key-field value with O(1). Redis also provide special commands that support synchronized data access. For example, BRPOP takes keys of List structures (one or more) as parameters and an integer number to specify the timeout in seconds. The command checks the specified lists in the same order given to the command and removes and returns the last element on that list. If all the lists are empty, the command blocks the current connection and waits for the amount of time specified by the timeout parameter for any other user connection that may be inserted to one of the lists before it release the connection and return a value to the client

3. HBase NoSQL database

A distributed, fault-tolerant, and with high scalability column-store NoSQL database implemented on top of the Apache Hadoop Distributed File System (HDFS), HBase is an Apache open-source database that provides real-time store and retrieving ability to massive data is. The data in HBase is arranged logically into named indexed tables. HBase tables are stored as multidimensional sparsely maps with rows and columns, where rows include a sorting key and an arbitrary number of columns. Versioning is used in table cells. When cells are added to HBase, HBase assigns a timestamp to them that is used to identify the version of that particular cell. For the same row key, many versions of a specific column might exist for that column. Column family and column name are assigned to each cell so that software can always tell what types of data item a particular set of cells contains. The content of a cell is an unbroken array of bytes that is uniquely recognized by the following combinations: Table + Row-Key + Column-Family: Column + Timestamp[9, 16].

A byte array, which also acts as the database's primary key, is used to sort the rows of the table

4. Experiments tool setup

4.1. Yahoo Cloud Service Benchmarking tool

We will use the Yahoo Cloud Service Benchmarking (YCSB) as the database performance evaluation tool. YCSB was created in 2010 by the research department at Yahoo. The task was to develop a tool that provides the ability to test and compare performance over the service provided by the cloud. Later, this tool becomes widely used by application developers to test database systems. In addition, this test can help during the decisions making to select the system to be used in the project. Figure 1 shows the tool architecture[6]. YCSB is developed using the

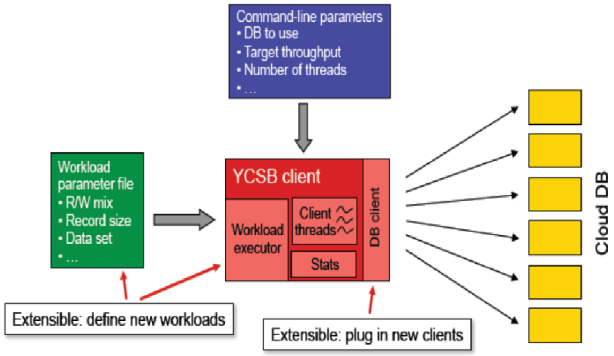


Figure 1. Architecture of YCSB.

Java programming language as an open-source project [10]. The code can be compiled with Maven and worked as a command line base. The tool support variety of NoSQL databases. The test is done by specifying the workload to be used. A workload can determine the number of operations and the types of these operations (Read, Write, and Update). There is a set of predefined workloads provided with tools default source code; we will use these workloads in this work, denoting them as (Load A, Load B, Load C, Load D, Load E, Load F). The test is done in two steps: the Load command and the Run command. The database connection information can be provided as a parameter to the tool with the Run and the Load command.

4.2. Hardware and software specifications

Table 1 below shows the system specification we used for this work.

We conduct the test using six workloads. We recorded the result by changing the number of threads used in the test. For each test, we build a chart that

Table 1. Hardware and software specifications.

System	
Operating System	Window 10 64 bit
Memory (RAM)	8GB
CPU	Intel Core i5-1135G7 4 x 2.4 - 4.2 GHz
Software	
Yahoo Cloud Service Benchmarking Redis	Version ycsb-0.17.0 Version 6.2.6
HBase	Version 2.4.9
Maven	Version apache-maven-3.8.4z

shows the recorded performance (throughput measured by operation per second) for both databases while changing the number of used threads. The number of threads can be determined in practice according to the application. The result for each workload is shown below:

4.3. Load A

In this workload, the tool divides the total operation into 50% read, and 50% write operation. Thus this workload can be considered heavy in terms of updates. The result is shown in Figure 2 below. We notified that the HBase started to give better performance when we increased the thread from six to seven threads with this load. However, we got a similar performance gap with more than seven threads. Thus, this load shows better performance for HBase in comparison to Redis.

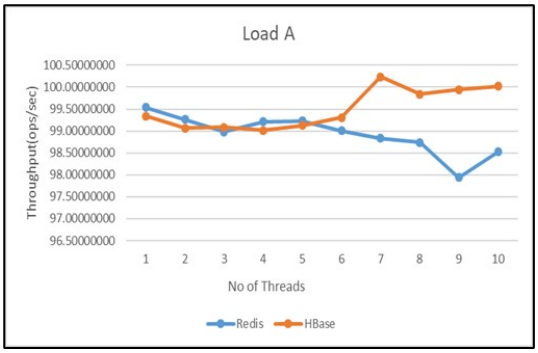


Figure 2. Load A.

4.4. Load B

The read operation takes 95% of the total operations in this workload. Thus we can denote this workload as reading heavy test. The max recorded throughput for

Redis and HBase is 99.54 100.33 milliseconds, respectively. The result is shown in Figure 3. Again, the HBase performs better than Redis with eight or more threads. Redis has no notifiable change during all the threads experiment.

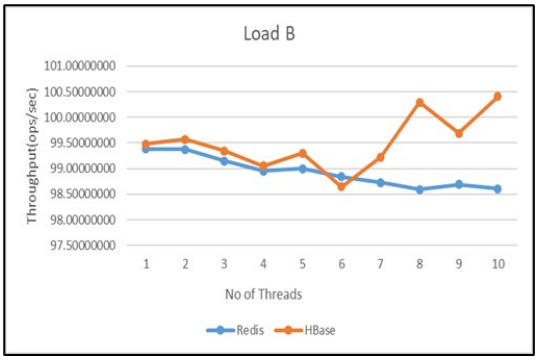


Figure 3. Load B.

4.5. Load C

This workload consists only of read operations and can be used to test the database when the application is critical to data retrieval, and there is no rapid insertion or update operation that can affect the software. The max recorded throughput for Redis and HBase is 99.36 and 100.39 milliseconds, respectively. The result is shown in Figure 4.

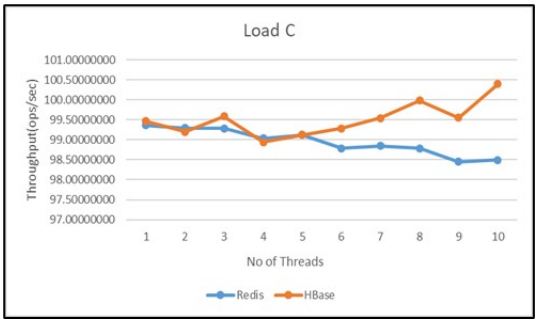


Figure 4. Load C.

4.6. Load D

This load contains only 5% insert operation with 95% read operations. The read operations are done on the data that was inserted recently. The max recorded throughput for Redis and HBase is 99.48 100.61 milliseconds, respectively. Figure 5

shows the Load D result. HBase shows better performance with increasing the number of threads.

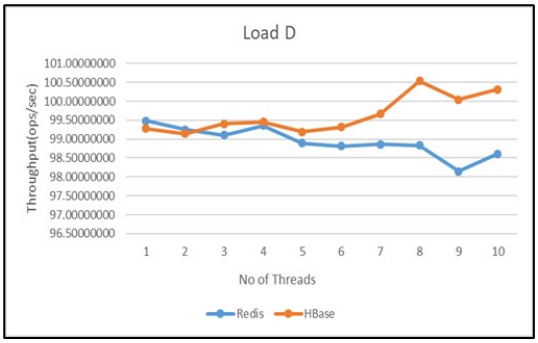


Figure 5. Load D.

4.7. Load E

95% of the time is spent scanning, and just 5% is spent inserting. It is scan for a short number of records rather than a single one. Figure 6 shows the result comparison for both databases. The max recorded throughput for Redis and HBase is 99.48 100.87 milliseconds. Both databases show similar performance till we use seven threads. However, the performance gap after seven or more threads was smaller compared to the gap we got with the other tests. Again the HBase was slightly better than Redis for this test.

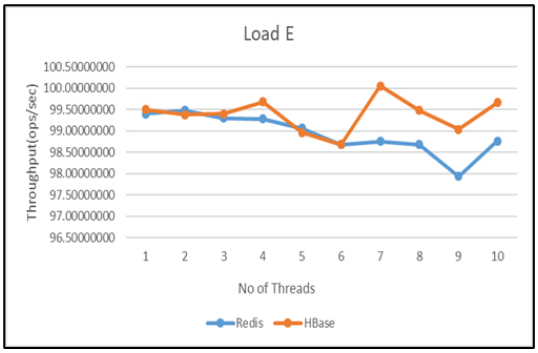


Figure 6. Load E.

4.8. Load F

This load simulates the situation when the application retrieves the data from the database, updates it, and then stores it back in the database. Figure 7 shows the

result for load F.

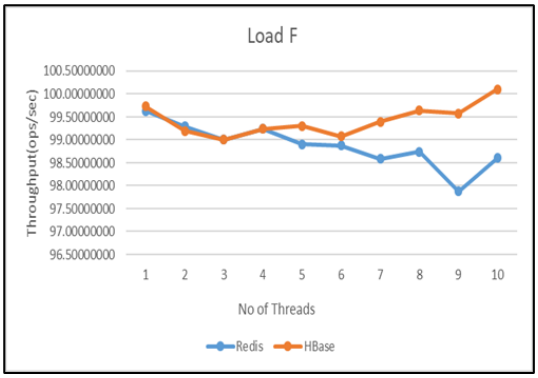


Figure 7. Load F.

5. Conclusion

Applications programmers may choose between SQL and NoSQL databases. Although their antiquity, SQL databases are still popular among programmers and web designers alike. The NoSQL database systems have become a good alternative to relational databases in some applications during the last decade. As they provide better scalability and schema-less structure, what can make software project development faster and easier. This advantage and popularity lead to the introduction of many NoSQL database systems. However, each may provide some features and miss some others that are provided by another system. Thus, the selection between the available NoSQL databases becomes more complex and needs a comparison between the candidate systems. We use the Yahoo Cloud Service Benchmarking tool to compare two popular NoSQL databases. We used the default workload provided by the tool, and we re-conducted the test using a different number of threads every time (1 to 10 threads). The results show that both databases have almost similar performance when fewer threads are used (less than 7). However, when we increase the number of used threads, the HBase shows higher throughput in compare to Redis.

References

[1] D. Abadi: *Column Stores for Wide and Sparse Data.* In: Feb. 2007, pp. 292–297.

[2] A. V. M. Alzaidi: *Trip Planning Algorithm For Gtfs Data With Nosql Structure To Improve The Performance*, Journal of Theoretical and Applied Information Technology Vol.99. No (10 31st May 2021 May 2021), pp. 2290–2300.

[3] E. Anderson, X. Li, M. Shah, J. Tucek, J. Wylie: *What consistency does your key-value store actually provide?*, HP Laboratories Technical Report (Feb. 2010).

[4] B. Atikoglu, Y. Xu, E. Frachtenberg, S. Jiang, M. Paleczny: *Workload analysis of a large-scale key-value store*, Sigmetrics Performance Evaluation Review - SIGMETRICS 40 (Feb. 2012), doi: https://doi.org/10.1145/2318857.2254766.

[5] R. Cattell: *Scalable SQL and NoSQL data stores*, SIGMOD Record 39 (Feb. 2010), pp. 12–27, doi: https://doi.org/10.1145/1978915.1978919.

[6] C. Chakraborttii: *Performance Evaluation of NoSQL Systems Using Yahoo Cloud Serving Benchmarking Tool*, in: Feb. 2015.

[7] C. Chasseur, Y. Li, J. M. Patel: *Enabling JSON Document Stores in Relational Systems.* In.

[8] B. Cooper, A. Silberstein, E. Tam, R. Ramakrishnan, R. Sears: *Benchmarking cloud serving systems with YCSB*, in: Feb. 2010, pp. 143–154, doi: https://doi.org/10.1145/1807128.1807152.

[9] L. George: *HBase: The Definitive Guide: Random Access to Your Planet-Size Data*, 1st, O'Reilly Media, Inc.: Sebastopol, CA, USA, 2011.

[10] https://github.com/brianfrankcooper/YCSB.: *YCSB*, in.

[11] V. Kacholia, S. Pandit, S. Chakrabarti, S. Sudarshan, R. Desai, H. Karambelkar: *Bidirectional Expansion For Keyword Search on Graph Databases.* In: vol. 2, Feb. 2005, pp. 505–516.

[12] H. Khazaei, M. Fokaefs, S. Zareian, N. Beigi, B. Ramprasad, M. Shtern, P. Gaikwad, M. Litoiu: *How do I choose the right NoSQL solution? A comprehensive theoretical and experimental survey*, Journal of Big Data and Information Analytics (BDIA) 2 (Oct. 2015), doi: https://doi.org/10.3934/bdia.2016004.

[13] K. Ma, A. Abraham: *Toward lightweight transparent data middleware in support of document stores*, in: 2013, pp. 253–257, doi: https://doi.org/10.1109/WICT.2013.7113144.

[14] T. Madushanka, L. Mendis, D. Liyanage, C. Kumarasinghe: *Performance Comparison of NoSQL Databases in Pseudo Distributed Mode: Cassandra, MongoDB & Redis* (Feb. 2015).

[15] A. Oussous, F.-Z. Benjelloun, A. A. Lahcen, S. Belfkih: *Comparison and Classification of NoSQL Databases for Big Data*, in: Feb. 2015.

[16] M. N. Vora: *Hadoop-HBase for large-scale data*, Proceedings of 2011 International Conference on Computer Science and Network Technology 1 (2011), pp. 601–605.

Annales Mathematicae et Informaticae
56 (2022) pp. 10–21
DOI: https://doi.org/10.33039/ami.2022.12.005
URL: https://ami.uni-eszterhazy.hu

A psychometric approach to email authorship assertion in an organization

Prathamesh Berde[a], **Manoj Kumar**[b], **C.S.R.C. Murthy**[b], **Lalit Dagre**[b], **Seervi Tejaram**[b]

[a]Homi Bhabha National Institute Mumbai, India
prathameshb@hbni.ac.in

[b]Bhabha Atomic Research Centre, Mumbai India
kmanoj@barc.gov.in
murthy@barc.gov.in
lalitd@barc.gov.in
tejas@barc.gov.in

Abstract. Email services have become an integral aspect of modern communication. Emails can be transmitted digitally without the adequate authentication of the sender. As a result, there has been a considerable surge in security threats coming from email communication, such as phishing, spear phishing, whaling, and malware deposition through emails where recipients can be duped into acting. Authorship assertion of the sender can prevent several security issues, particularly in an organizational setting where an employee's trust can be compromised by faking an email from a colleague or senior without exposing any specific system weakness. A psychometric approach to determining the authorship of an email in an organization is proposed in this research. Machine learning (ML) models have been developed using four classification algorithms. The performance of these ML models has been compared.

Keywords: authorship, personality, machine learning, psychometric features

1. Introduction

The Internet has become an integral part of our life. In modern-day communication, the predominant mode of communication on the internet is Email. Email service impales very deep into private networks and intranet of organizations, thereby allowing attackers to deploy the exploits far into organizations' networks. Hence,

Submitted: October 26, 2022
Accepted: December 3, 2022
Published online: December 28, 2022

the security of email service is one of the major tasks in an organization. One of the prominent attacks on email is the social engineering attack. The knack of influencing the people to divulge sensitive information of some other action is known as social engineering and the process of doing it is called the social engineering attack [13]. In some of the modern-day social engineering attacks against one victim or a small group thereof, the attackers research their targets to design phishing emails specific for each victim. The emails appear to be coming from a trusted colleague/party and prompt the recipient to follow the directions inside. By impersonating trusted email senders through meticulously crafted messages, attackers trick the receivers to act on that email and launch malware. Such an attack is mostly used as a platform for injecting malware into interior parts of an organization such as the Intranet. Attacks involve targeting individuals from organizations by maneuvering them to promulgate misleading information to varied interests and valuable and sensitive data that may intrigue cybercriminals without exploiting a specific vulnerability. As discussed in [1], emails can transmit information digitally without authenticating the person who writes the text and could be used by criminals for malicious intentions. Authorship assertion of such emails becomes necessary in an organization.

Alhijawi et al. [1] surveyed some of the possible techniques for authorship attribution. They carried out the authorship analysis technique to satisfy the objective. Authorship Identification, similarity detection, and characterization were its three main perspectives. Their survey showed the use of stylometric features for authorship identification. The features were classified into four categories namely lexical, character, syntactic and semantic. Lexical features included token-based, vocabulary richness, word frequencies, word n-grams, errors, character features included character types, character n-grams, etc. Syntactic and semantic features included the parts of speech and semantic dependencies. Some of the datasets in the research were email datasets, online text data sets, source code data sets, etc. Yet, it is observed that the features used in this research may not be invariant as the context of the writing changes.

One of the approaches in this field is the classification of authors' emails based on their representation of text to vectors [4]. Here, they used the word2vec to generate the word embedding and extract the features of the author's writing style from their text writing. Multi-layer Perceptron classifier and the back-propagation learning algorithm were used for classification. They used the PAN12 free fiction collection data corpus written in English. A cluster-based classification model for email authorship identification was also used [15]. Stylometric features like punctuation used at the end of the emails, the tendency of the user to start the emails with the capitalized letters, punctuation after the greetings and farewells, etc were used for classification. The dataset used for their analysis was the Enron email dataset.

One of the other works in this field, carried the authorship identification for short online messages [5] using Supervised Learning and n-gram analysis. Enron email dataset was used for their analysis. One of the works used an approach of

Unsupervised Clustering for authorship identification [14] for email forensics where they classified emails initially using unsupervised clustering and then identified the stylometric features in the clusters. They used the Hierarchical Clustering and Multidimensional scaling approach of Unsupervised Clustering for authorship identification. They also used the Enron email corpus data set for their experimentation.

The motive behind carrying out the work presented in our paper was to develop classification models of known authors in an organization so that the impersonated emails claiming to be coming from these authors could be asserted. Hence, this work emphasized developing models that assert authorship of an email in an organization using Machine Learning algorithms for known email authors.

The remainder of the paper is organized as follows. Section 2 introduces the methodology used for authorship assertion. Section 3 presents the details of feature extraction and training of the ML classifier. Validation of feature extraction models is discussed in Section 4. An analysis and comparison of performance metrics of different ML models are discussed in Section 5.

2. Methodology

The proposed approach to email authorship assertion in this paper is based on the fact that personality is a stable and invariant aspect of an individual [9] and the most relevant differences/traits are encoded in the language written [3]. Using these characteristics of the personality and language (extracted from emails), the problem of authorship assertion is transformed into a classification problem. To formulate the classifier, the following are needed:

2.1. Evaluation of personality score from the questionnaire

Personality is the characteristic pattern of those sensory, perceptual and cognitive systems within an individual that determines his unique behavior in his environment [2]. The Big Five Personality Model is one of the most widely used models of personality. This model is also known as the five-factor model or the OCEAN model which is based on five personality dimensions i.e. Openness, Conscientiousness, Extroversion, Agreeableness, and Neuroticism [9]. Volunteering authors undergo a personality assessment test and personality scores are generated. The scores are based on the International Personality Item Pool proxy for the NEO Personality Inventory-Revised (NEO PI-R) questionnaire [6]. NEO PI-R is considered by many psychologists for measuring the dimensions within the Big Five Personality Model.

Statistics about the personality dimensions evaluated from the questionnaire have been given in Table 1.

Table 1. Statistics of personality scores of users after NEO PI R personality questionnaire.

Dimension	**Mean**	**Standard Deviation**
Neuroticism	48.17	30.41
Openness	30.17	23.75
Agreeableness	62.53	22.23
Extroversion	43.74	28.56
Conscientiousness	67.03	21.13

2.2. Extraction of word category lexica from emails

Various word categories are described in the word category lexica of the content-coded dictionary of the packages provided in [7, 20] available on LIWC [17]. Word count corresponding to various parts of speech (POS) categories like articles, conjunctions, etc. using Spacy [10] in the Python programming language is extracted from the emails. The word count corresponding to each word category like positive, negative words, sadness, achievements, etc. in the dictionary using the Empath [7] package in the Python programming language is derived from the emails. The word count corresponding to each category is appended to a column vector for an email.

2.3. Feature vector extraction for classifier and authorship assertion

The feature vector for the classifier consists of a score of personality corresponding to the personality dimension in the five-factor model. To extract the personality dimension scores, a regression model may be used. The regression model estimates the personality score using the correlation of personality score evaluated from the authors' questionnaire and their corresponding emails' column vectors as discussed in Section 2.2. The classifiers are trained using features of old emails and subsequently used for authorship assertion of new emails they claim to be coming from.

For implementing regression models to extract personality scores, Linear Regression, Support Vector Regression (SVR), Regression Trees, and Neural Networks have been used. For the classification of emails in the last stage, Logistic regression, Support Vector Machine (SVM), Neural Networks, and Naive Bayes have been used. All the algorithms have been implemented in Python 3 using the modules of Scikit-learn [16].

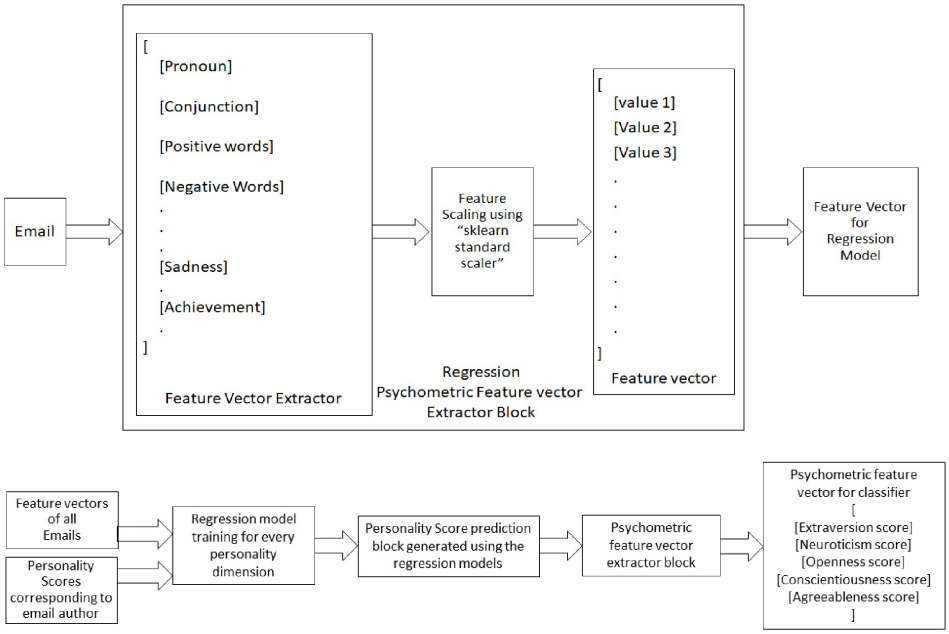


Figure 1. Psychometric feature vector extraction.

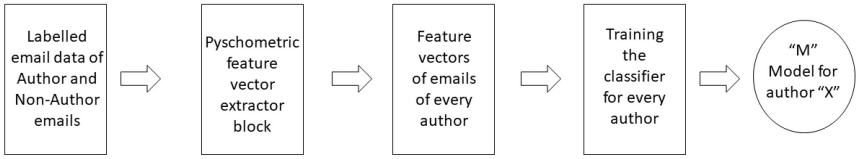


Figure 2. Training of classifier using Psychometric Features.

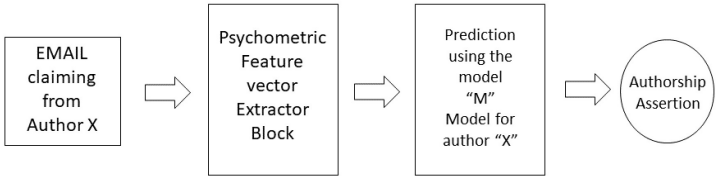


Figure 3. Classification Stage.

3. Implementation

For authorship assertion in the organization, the experiment was conducted on a limited set of 18 users. These users have volunteered, given consent to use their

past emails, and answered the questionnaire for personality dimensions [6]. We tried to develop the author-specific models to analyze if the email had been sent by him or not. Out of all the authors who volunteered, the classifier analysis of the 5 authors who had the highest number of emails is discussed in this paper.

3.1. Data preparation and pre-processing

The first stage of the implementation was data preparation. In the data preparation, a data frame was prepared for analysis of the data. The sent emails of the authors were used. The sent emails of the users had been collected for the past 1 year and only those emails were considered in which the author had started the conversation. The forwarded and replied emails were not considered in the analysis. Using the standard python programming language libraries, we pre-process the data and extract the text corresponding to the email bodies. The email body content for every email was separated after extracting the message in the email and the signatures from the emails were stripped off as discussed in [8, 21]. Emails were appended in a data frame. Now corresponding to every processed email, the score of personality dimension which had been collected from the questionnaire of the corresponding user was assigned.

3.2. Feature extraction and training

Regression techniques were used to relate word categories with authors' personality scores. As shown in Fig. 1, the scores for each personality dimension of the author were assigned and the counts corresponding to each lexical category of word category lexica were extracted from emails as inputs to the regression model as discussed in Section 2.2. Regression algorithms were used to fit a curve between independent factors i.e. the lexical categories and the regressand i.e. Extraversion, Neuroticism, Openness, Agreeableness, and Conscientiousness. The following steps were involved.

- Features were extracted by obtaining word count corresponding to various parts of speech and the word count corresponding to every lexical category for every email in the dataset using respective packages in python programming language as discussed in Section 2.2.
- Feature scaling was performed as the features varied in terms of what they represent. Some algorithms are invariant to feature scaling while some are not.
- Once the features were scaled, the regression models were trained using the regression algorithms specified in Section 2.3.
- Regression Algorithms like SVR and Neural Networks used various hyperparameters while training. Optimum hyperparameters for improving performance were chosen by hyperparameter tuning.

- After the Hyperparameters had been optimized, results and performance of the machine learning algorithms were compared and the model with the best performance evaluated using standard metrics[11] in Regression was chosen for the prediction of the score for every personality dimension.

- To verify whether the regression model correctly prepared data for the classifier and whether the features used for machine learning were sufficient to be used for a classifier, clustering analysis using K-means clustering and the Density-based spatial clustering of applications with noise (DBSCAN) clustering algorithm was performed and the goodness of the cluster was analyzed using standard metrics[18, 19].

3.3. Email classification for authorship assertion

After the generation of the regression model, a data set of emails was prepared for every author. In a particular data set, we selected all the emails which belonged to that author, and we randomly selected an equal number of emails that did not belong to that author. Then, for every email, we extracted the personality scores using the regression model generated in the previous step. As shown in Fig. 2, we extracted the feature vectors and modeled the author-specific classifier. The following steps were involved in this stage.

- An author and his email data for a year were collected. Then, we randomly selected the same number of emails as the author from data that did not belong to the selected author and labeled them correctly.

- The personality scores to each of these emails from the dataset were extracted using the regression models for each dimension in the previous stage and a feature vector matrix was derived which was followed by feature scaling.

- Once the features were scaled, the classification models were trained using the classification algorithms specified in Section 2.3 and the hyperparameters were optimized.

- After the Hyperparameters were optimized, we compared the results and performance of the machine learning algorithms and chose the algorithm with the best performance after analyzing using standard metrics for classification[11], and saved the model for use in the classification of email whether it belongs to the specified author.

As shown in Fig. 3, when a new email was received we first extracted the features using the regression model i.e the personality scores, prepared the feature vector matrix, and then predicted the class of this vector using the classification model of the author it claims to be coming from.

4. Validation of regression models

The performance of regression algorithms is given in Table 2. It is evident from the results that SVR outperformed any other regression algorithm for this data. It was also evident in the literature survey that kernelized regression algorithms like SVR have performed better than other algorithms. R^2 value for SVR was higher than other regression algorithms. The Mean Absolute Error (MAE) percentage was also relatively less compared to other regression algorithms. The decision to use these metrics for selecting the models was based on the facts published in the literature survey [12]. It is also to be noted that R^2 values mean the percent of explained variance on the dependant variable. So in our experiments when we tried to analyze the impact of a certain limited number of variables on the human-related outcomes, it was very difficult to explain the majority of the variance.

Table 2. Performance of personality score prediction model for psychometric feature vector extraction.

	Neuroticism		**Openness**		**Agreeableness**		**Extraversion**		**Conscientiousness**	
Algorithm	R^2	% MAE	R^2	% MAE	R^2	% MAE	R^2	% MAE	R^2	% MAE
Linear regression	0.15	34.37	0.15	25.04	0.3	18.34	0.14	18.83	0.15	30.37
Support vector Regression	0.43	12.65	0.36	13.47	0.41	13.19	0.44	12.32	0.42	15.96
Decision tree regression	0.29	23.94	0.18	18.63	0.24	23.86	0.31	15.25	0.28	21.43
Neural Network regression	0.18	27.3	0.16	23.67	0.35	17.28	0.24	16.11	0.19	26.28

Table 3. Performance of clustering algorithms to analyze data separability.

	K-means Clustering		**DBSCAN Clustering**	
No. of users	3 users	5 users	3 users	5 users
Estimated clusters	-	-	4	6
Silhouette Coefficient	0.714	0.69	0.68	0.59
Homogeneity	0.886	0.782	0.77	0.697
Completeness	0.881	0.78	0.774	0.63
V- Measure	0.884	0.781	0.771	0.662

To verify whether the regression model correctly prepared data for the classifier and whether the features used for machine learning are sufficient to be used for a classifier we performed clustering analysis using K-means clustering and the DBSCAN clustering algorithm. To perform the clustering analysis, 5 volunteers out of 18 having the highest number of emails were considered. It was evident from the results shown in Table 3 of the clustering analysis that the SVM regression

model, has features sufficient to explain the variation in personality and can be used to derive the features for training the classifier and we can model a supervised classifier for the analysis of the same.

5. Results and discussion

Metrics like accuracy, f-score, sensitivity, specificity, training time, and prediction time were evaluated for the choice of the best models. It was desired that emails that do not appear to be coming from the author should be asserted correctly as such emails may create havoc if undetected. We chose to decide on the best model by comparing prediction accuracy, prediction time, and specificity. In this work, we were able to achieve accuracy which was in the range of 80-95% for authorship assertion. The features used relied on the personality dimensions of the five-factor model of personality. It was observed from the performance of classification algorithms shown in Table 4 that the Neural Network classifier and the SVM classifier have comparable performance considering the accuracy of the model trained using psychometric features. These two classification algorithms perform better than Naive Bayesian and the Logistic Regression classification algorithm. From the clustering analysis, we observed that although the data used for training the classifier was separable, it is not perfectly homogeneous i.e. each cluster did not have data points belonging to the same class label. The SVM algorithm implemented in the classifier used in this approach required two hyperparameters, C and γ along with kernel functions to separate the two classes using a hyperplane. Kernel functions only calculated the relationship between every pair of points as they are in a higher dimension. Parameter C traded off misclassification of training data points against decision surface while γ determined how much influence a single training datapoint has. Optimum choice of the kernel function, values of C and γ were predicted using hyperparameter tuning.

The neural network learned the nonlinear function approximator using the summation of weighted layers of neurons and their transformation at the output of each neuron using its activation function for two classes using the various hyperparameters and optimum hyperparameters were obtained by hyperparameter tuning. Neural networks required a higher training time as the initialization of weights was done according to standard method i.e. by initializing weights and bias of the complex neural network by random number generation and were optimized by error backpropagation using stochastic gradient descent solver after every iteration, although the prediction time was not much higher as the weights had been tuned during the training phase. Due to the above reasons, SVM and Neural Networks were able to fit and perform better than other algorithms on the nonlinear and not perfectly homogeneous data points used in this analysis.

In the training phase as well as the testing phase, no other classifier was as fast as the Naive Bayesian classifier (the value of this metric was 2-3 milliseconds) because training the Naive Bayes classifier required the calculation of the probability of individual classes and the class conditional probabilities. Also, optimization pro-

Table 4. Performance of classification algorithms.

user	algorithm	accuracy	f1_score	sensitivity	specificity	training time (in s)	prediction time (in ms)
USER 1	Logistic regression classifier	86.86	0.87	0.87	0.9	0.015	0.002
	SVM classifier	90.06	0.9	0.9	1	0.345	0.072
	Neural Network classifier	89.74	0.9	0.9	0.95	1.749	0.005
	Naive Bayes classifier	88.78	0.89	0.89	0.91	0.003	0.003
USER 2	Logistic regression classifier	86.33	0.86	0.86	0.83	0.02	0.003
	SVM classifier	89.45	0.89	0.91	0.97	0.362	0.079
	Neural Network classifier	94.92	0.95	0.95	0.96	0.869	0.005
	Naive Bayes classifier	90.63	0.9	0.89	0.82	0.003	0.003
USER 3	Logistic regression classifier	94.32	0.94	0.94	0.89	0.025	0.002
	SVM classifier	95.63	0.96	0.95	0.92	0.127	0.036
	Neural Network classifier	94.76	0.95	0.94	0.9	0.605	0.005
	Naive Bayes classifier	95.63	0.96	0.95	0.92	0.003	0.003
USER 4	Logistic regression classifier	80.37	0.8	0.81	0.78	0.02	0.003
	SVM classifier	90.8	0.9	0.89	1	0.113	0.047
	Neural Network classifier	85.28	0.85	0.85	0.86	0.633	0.006
	Naive Bayes classifier	85.89	0.86	0.86	0.87	0.002	0.003
USER 5	Logistic regression classifier	84.81	0.85	0.85	0.86	0.02	0.003
	SVM classifier	87.97	0.88	0.87	0.99	0.114	0.047
	Neural Network classifier	92.41	0.92	0.92	0.99	0.625	0.006
	Naive Bayes classifier	82.91	0.83	0.84	0.73	0.002	0.003

cedures did not require the calculation of coefficients. Additionally, the algorithm assumes all features to be independent, and hence parametric calculations can be done individually and faster.

The prediction using SVM is comparatively slower because before prediction SVM transforms the input vector to a higher dimensional feature vector. Additionally, SVM used kernel trick to reduce the computation time in high dimensional feature space. Prediction time using all the algorithms is comparable in a few microseconds. Another important aspect that we analyzed was specificity. Specificity determined the fraction of actual negative cases which got predicted correctly. In our data, actual negative cases were those emails that do not belong to that user. We observed that the SVM classifier outperformed other classifiers on this metric (the value of this metric existed between 0.9 and 1). Hence, the use of an SVM classifier to train the classification model using the psychometric features is recommended.

6. Conclusion

The proposed technique is based on the fact that a person's personality is a constant and stable quality that is represented in his language. The authorship assertion problem has been treated as a classification problem using these principles. To develop the classifier, a questionnaire to assess personality traits has been used, then

the extracted word category lexica from emails are used to develop the personality score prediction model, followed by feature vector extraction and training of classifiers. A comparison of models developed using four classification algorithms was conducted to evaluate and choose the best model for each author based on parameters like accuracy, specificity, prediction time, and so on. On these metrics, SVM and Neural Network classifiers outperformed others.

Although these models function commendably, there may be inconsistencies if the threat actor and the real sender have similar personalities. Another inconsistency may develop if the personality scores collected via the personality questionnaire have not been attempted truthfully, since this may represent misleading personality behavior in the scores, making the training of the regression model erroneous. The work can be improved in the future by defining a more comprehensive set of features and employing advanced machine learning models. Model boosting and bagging may also increase performance and the development of models.

Acknowledgement. We would like to express our sincere gratitude to the Head, Computer Division, BARC for providing us with the data. We would thank Shri Rohitashva Sharma for providing the necessary infrastructure and allowing us to carry out this work at HBNI Complex. We would also thank Shri Shankar for the support.

References

[1] B. Alhijawi, S. Hriez, A. Awajan: *Text-based Authorship Identification - A survey*, in: 2018 Fifth International Symposium on Innovation in Information and Communication Technology (ISIICT), 2018, 1–7.

[2] G. W. Allport: *Personality: A Psychological Interpretation.* (1937).

[3] G. W. Allport, H. S. Odbert: *Trait-names: A Psycho-Lexical Study.* Psychological monographs 47.1 (1936), p. i.

[4] N. E. Benzebouchi, N. Azizi, N. E. Hammami, D. Schwab, M. C. E. Khelaifia, M. Aldwairi: *Authors' Writing Styles Based Authorship Identification System Using the Text Representation Vector*, in: 2019 16th International Multi-Conference on Systems, Signals Devices (SSD), 2019, 371–376.

[5] M. L. Brocardo, I. Traore, S. Saad, I. Woungang: *Authorship Verification for Short Messages using Stylometry*, in: 2013 International Conference on Computer, Information and Telecommunication Systems (CITS), IEEE, 2013, 1–6.

[6] P. T. Costa Jr, R. R. McCrae: *The Revised NEO Personality Inventory (NEO-PI-R).* Sage Publications, Inc, 2008.

[7] E. Fast, B. Chen, M. S. Bernstein: *Empath: Understanding topic signals in large-scale text*, in: Proceedings of the 2016 CHI conference on human factors in computing systems, 2016, pp. 4647–4657.

[8] H. Gascon, S. Ullrich, B. Stritter, K. Rieck: *Reading Between the Lines: Content-Agnostic Detection of Spear-Phishing Emails*, in: Research in Attacks, Intrusions, and Defenses, Springer International Publishing, Springer, Cham, 2018, 69–91, isbn: 978-3-030-00470-5.

[9] L. R. Goldberg: *An Alternative "Description of Personality": The Big-Five factor structure.* Journal of Personality and Social Psychology 59.6 (1990), p. 1216.

[10] M. Honnibal, I. Montani, S. Van Landeghem, A. Boyd: *spaCy: Industrial-strength Natural Language Processing in Python*, https://doi.org/10.5281/zenodo.1212303, 2020, doi: {10.5281/zenodo.1212303}.

[11] Joshi, Ameet V: *Machine Learning and Artificial Intelligence*, Springer, 2020.

[12] F. Mairesse, M. A. Walker, M. R. Mehl, R. K. Moore: *Using Linguistic Cues for the Automatic Recognition of Personality in Conversation and Text.* Journal of Artificial Intelligence Research 30 (2007), 457–500.

[13] F. Mouton, L. Leenen, H. S. Venter: *Social Engineering Attack Examples, Templates and Scenarios*, Computers & Security 59 (2016), pp. 186–209.

[14] S. Nirkhi, R. Dharaskar, V. Thakare: *Authorship Verification of Online Messages for Forensic Investigation*, Procedia Computer Science 78 (2016), 1st International Conference on Information Security & Privacy 2015, 640–645, issn: 1877-0509, doi: {https://doi.org/10.1016/j.procs.2016.02.111}, url: %7Bhttp://www.sciencedirect.com/science/article/pii/S1877050916001137%7D.

[15] S. Nizamani, N. Memon: *CEAI: CCM-based email authorship identification model*, Egyptian Informatics Journal 14.3 (2013), pp. 239–249, issn: 1110-8665, doi: https://doi.org/10.1016/j.eij.2013.10.001, url: http://www.sciencedirect.com/science/article/pii/S111086651300039X.

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

[17] J. W. Pennebaker, R. L. Boyd, K. Jordan, K. Blackburn: *The Development and Psychometric Properties of LIWC2015*, http://liwc.app/, 2015.

[18] Rosenberg, Andrew and Hirschberg, Julia: *V-measure: A Conditional Entropy-Based External Cluster Evaluation Measure*, in: Proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning (EMNLP-CoNLL), 2007, 410–420.

[19] P. J. Rousseeuw: *Silhouettes: a graphical aid to the interpretation and validation of cluster analysis*, Journal of computational and applied mathematics 20 (1987), pp. 53–65.

[20] Tausczik, Yla R and Pennebaker, James W: *The Psychological Meaning of Words: LIWC and Computerized Text Analysis Methods*, Journal of Language and Social Psychology 29.1 (2010), 24–54.

[21] R. Verma, N. Shashidhar, N. Hossain: *Detecting Phishing Emails the Natural Language Way*, in: Computer Security – ESORICS 2012, Springer Berlin Heidelberg, 2012, 824–841, isbn: 978-3-642-33167-1.

Annales Mathematicae et Informaticae
56 (2022) pp. 22–35
DOI: https://doi.org/10.33039/ami.2022.12.001
URL: https://ami.uni-eszterhazy.hu

Enhancing Hungarian students' English language skills on the basis of literary texts in the three-dimensional space*

István Károly Boda[a], **Erzsébet Tóth**[b], **László T. Nagy**[a]

[a]Department of Mathematics and Informatics,
Debrecen Reformed Theological University
boda.istvan@drhe.hu, t.nagy.laszlo@drhe.hu

[b]Department of Data Science and Visualization, University of Debrecen
toth.erzsebet@inf.unideb.hu

Abstract. In our paper we introduce a bilingual language learning material developed in the framework of the so-called three dimensional virtual library model (3DVLM). This model inspired by the history and organization of the famous ancient Library of Alexandria forms the basis of the virtual library project which started about eight years ago as part of the Cognitive Infocommunications (CogInfoCom) research. The current version of the 3DVLM uses the excellent 3D features of the MaxWhere Seminar System which makes it suitable for both individual learning and classroom use. In the following, we would like to introduce first the basic framework of our development, then describe in detail the data structure and organization of the developed bilingual language learning material. The basic idea of the material is to present selected phrases and contexts from classical literary works in English and from their parallel translations in Hungarian in order to improve both the language skills and background knowledge of Hungarian language learners at an advanced level. We found that using web technology was especially useful for developing the language learning material and the developed hypertext structure formed a scale-free network of interconnected nodes.

Keywords: second language learning, three-dimensional virtual library model (3DVLM), MaxWhere Seminar System, bilingual language learning material

AMS Subject Classification: 68U05, 68U35, 68T05, 91E10, 91E40

*This research has been supported by Virtual Reality Laboratory, Qos-HPC-IoT Laboratory and project TKP2021 NKTA-34 of the University of Debrecen, Hungary.

Submitted: November 5, 2022
Accepted: December 3, 2022
Published online: December 28, 2022

1. Introduction

In the year 2013 a *virtual library project* was initiated as part of the cognitive infocommunications (CogInfoCom) research [2, 3]. From the beginning, we have laid great stress on the mapping and visualization of the library content in the virtual 3D space the characteristics of which have been thoroughly investigated and analyzed by a lot of studies. We found especially useful for our project the presentation of virtual buildings in the 3D space [19, 27], the use of 3D VR as an effective virtual learning environment [20, 21], and the psychological aspects of the 3D environment [5, 6] but the number of such investigations is substantially increasing [15, 16]. The virtual library project was originally intended to bring together, arrange and show relevant verbal and multimedia materials in the 3D virtual space about the Great Library of Alexandria and Greek literary texts in English (e.g. preprocessed content about the work and life of Callimachus, English versions of chosen literary texts of remarkable ancient writers and poets etc.) [7, 9, 12], but later we significantly expanded the content of the virtual library in order that we can meet the requirements of the potential language learners. Though we think that the 3DVLM can be developed for different applications and purposes, *language learning* has seemed to be the most useful application of the virtual library material [10, 11] because, among others, of the increasing significance of the advanced English language competence and skills in the so-called information society. Moreover, the basic concept of the virtual library project includes to convey the message of ancient and classical cultures to the present-day culture through literature and we are convinced that with a carefully elaborated way and methodology the eternal values and thoughts of classical literary works can be precisely and eloquently expressed for the young members of the generations CE [15].

The current implementation of the 3D virtual library model exploits the spectacular 3D features of the MaxWhere Seminar System [26] especially because the arranged web browsers (called smartboards) fully support web technology and therefore enable the hypertext-based implementation of the basic concepts of the 3DVLM [8, 13, 15, 17].

In the following section we give an overview on the basic concepts and overall organization of the 3DVLM as a virtual learning environment where the selected and carefully preprocessed library content of the knowledge base of the virtual library will be presented for the potential language learners.

2. A brief overview of the 3DVLM as a virtual learning environment

As discussed before, the current implementation of the 3DVLM uses the innovative and spectacular 3D features of the MaxWhere Seminar System. We emphasize primarily the embedded *smartboards* in a selected ready-made 3D virtual space where the core content (e.g. texts about Callimachus or the Library of Alexandria, selected

parts of classical literary works etc.) and various navigation devices (thesaurus, index, concordance map, reference etc. *pages*) of the virtual library [13, 15, 17] can be displayed. A number of excellent and well-designed 3D virtual spaces can be found on the MaxWhere site [26] and they can be applied to almost every context, although each space shows its distinguished and unique characteristics. In our previous publications [13, 14, 16, 17] we selected the *3D Castle* virtual space for the presentation and arrangement of the virtual library content. But, owing to the flexibility of the 3DVLM, we can utilize other 3D spaces as well. Therefore we chose the *3D Library* virtual space for the new implementation of the virtual library model which provides a lot of smartboards in a virtual two-storey library building. In the following, we are going to show some screenshots and explanatory notes so as to illustrate how to have easy access to the preprocessed verbal and multimedia content in the 3D Library space.

Let us use the navigation page as a starting point [15, 17] (Fig. 1).

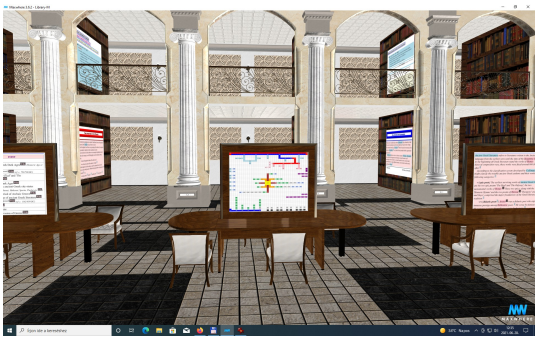


Figure 1. The *navigation page* of the virtual library content placed on the ground floor in the MaxWhere 3D Library space.

In the foreground of Fig. 1 there are three smartboards which jointly form an "information desk" of the 3D virtual library. These browser windows provide "smart" access to the main navigation devices of the virtual library:

- the *navigation page* is placed at the centre of the image;
- on the left side we can find a small part of the page providing a *timeline* of some historical events of the ancient era;
- on the right side a part of the *category page* [17] can be recognized which involves explanations of the main classification categories and presents their hierarchical structure.

In the background of the screenshot shown in Fig. 1 we can see some additional smartboards. Based on the content they contain we can distinguish two different types as follows:

- the smartboards located on the ground floor of the 3D library (the so-called *main cabinets*) show the core content of the virtual library including primary texts about Callimachus, the ancient Library of Alexandria etc. as well as selected parts of literary texts;
- the smartboards located on the first floor of the 3D library show, among others, the so-called *thesaurus pages* of the virtual library. These pages are intended to present additional linguistic knowledge which has been organized around certain keywords and collocations selected from the texts of the cabinets, and represented by a number of concordances or quotations which contain at least one of the keywords in the given collocation pattern.

Note that the developed bilingual language learning material can be considered as a supporting device for the language learners which contains designated keywords and selected contexts from classical and modern literary works. Therefore its place in the virtual 3D Library environment can be either on the ground floor (among literary texts which can directly refer to the material) or on the first floor (among the thesaurus pages which support e.g. vocabulary building).

The main function of the information desk is to enable the users to access relevant information, hence we located the content of the navigation pages also on the wall of the 3D library (see Fig. 2).

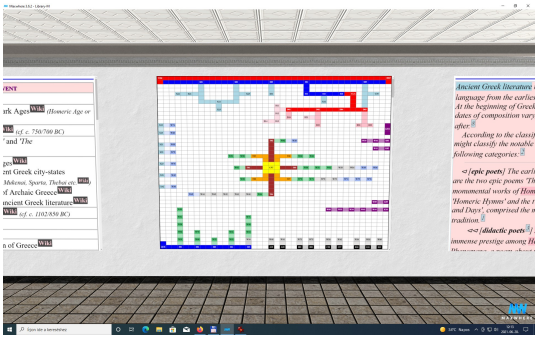


Figure 2. Three *navigation pages* of the virtual library placed on the wall in the MaxWhere 3D Library space.

The content of some of the main cabinets is organized around selected primary texts about the life and work of Callimachus (including the Pinakes, the ancient Library of Alexandria, the works of Callimachus etc. [13, 15–17] which, as we mentioned before, can be discovered on the ground floor of the 3D library just behind the information desk. The primary text about the ancient Library of Alexandria, and that about Callimachus can be observed in Fig. 3.

From a different view we can see the primary text about the Pinakes as well (Fig. 4).

For those who would like to see the hypertext representation of the library content we have mentioned above, the current content of the virtual library project

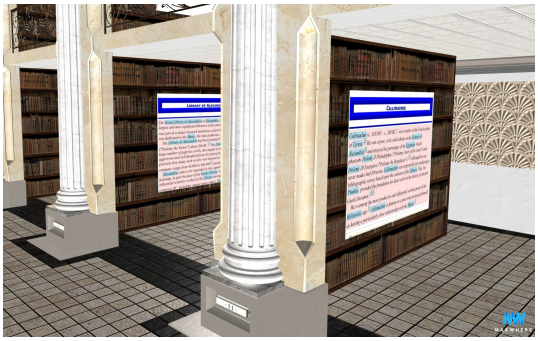


Figure 3. *Cabinets* which include primary texts about the Library of Alexandria and Callimachus placed on the ground floor in the MaxWhere 3D Library space.

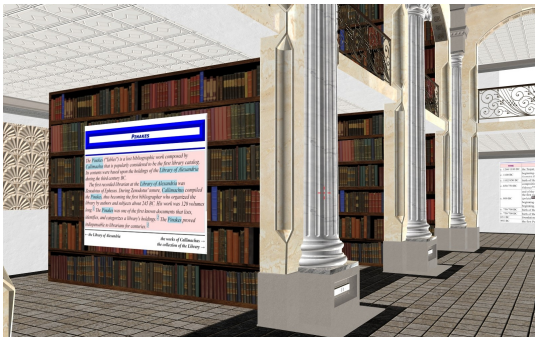


Figure 4. The *cabinet* which shows the primary text about the Pinakes placed on the ground floor in the MaxWhere 3D Library space.

can be accessed through the internet [23].

3. Introduction of a bilingual learning material for language learners

In the following, we would like to introduce the latest development of our virtual library project. We prepared a *bilingual language learning material* [25] aimed especially at Hungarian students who have an advanced level of English language proficiency (and who have great interest in literature as well). The basic idea of the material is to present carefully selected passages from literary works along with their parallel translations and organize them with the intention to prepare a more or less scale-free network of interconnected nodes in order to provide an efficient learning environment for language learners.

We'll have a ***swashing*** *and a* ***martial*** *outside* (I.3.120)

where the adjectives *'swashing'* and *'martial'* have several synonyms as well as rich connotations which we thought were worth elaborating. So we gathered two separate groups of semantically related words named as Part 1 and Part 2, respectively. Each of the groups had more than 60 items, e.g.

loud, noisy; hoarse, rough, harsh; ...; hectoring, boastful, cocky; *swaggering*, **swashing**, swashbuckling, square-jawed; ...; disdainful, contemptuous, scornful (Part 1)

active, energetic, vigorous, dynamic, alert; ...; **martial**, soldierly, militant, combative; *aggressive*, bellicose, belligerent, quarrelsome; ...; relentless, implacable (Part 2)

These words have been considered as keywords and *the primary aim of the developed bilingual learning material is to help language learners to enhance their vocabulary* as well as their language skills by learning these words and their contexts.

Although we gave Hungarian translations of the listed English words, we added selected bilingual phrases and sentences (either alone or with a broader context) to the material in order that the possible language learners could deepen, interconnect and then memorize the whole content. Moreover, we organized the content of the material by devising an inner hyperlink structure where

- the keywords serve as nodes and
- the selected contexts of the keywords contain hypertext links to the keywords that occurred in the contexts.

Metaphorically speaking, **we considered the bilingual learning material as a hypertext-based model for the long-term memory of the language learners.**

We selected 20 literary works in English (both from the English literature and from the world literature in English translations) with their parallel Hungarian translations as sources for the selected contexts that contain at least one of the keywords to be learned. As for the bilingual phrases, the available dictionaries proved to be a rich source in addition to the texts of the selected literary works. In some cases we also provided sentence examples, but this option could be switched on or off depending on the demands of the users of the learning material.

The literary works include English classics such as William Shakespeare's *As You Like It*, Jane Austen's *Pride and Prejudice*, Charlotte Bronte's *Jane Eyre*, Sir Arthur Conan Doyle's *The Adventures of Sherlock Holmes* etc. Works from the world literature in English translations include Victor Hugo's *Les Miserables*, Rafael Sabatini's *Captain Blood*, Leo Tolstoy's *War and Peace* etc. We would like to add some present-day literature works, too; so we selected short passages from J. K. Rowling's famous Harry Potter series, Stephenie Meyer's Twilight saga etc.

4. The data and link structure of the bilingual learning material

As we mentioned above, we had gathered more than 120 keywords which formed separate nodes in the hypertext structure. We attached carefully selected bilingual phrases, sentences and contexts to almost all nodes. (Note that each context established an individual node as well). We were aware that in the contexts which came from literary works there could be unknown, rare or difficult words or phrases, so we added separate vocabulary entries to each context in a separate section called 'Comments'. We also added further vocabulary entries to every keyword that occurred in a specific context and then, in each vocabulary entry, established hypertext links from each keyword to the corresponding node. For example, in Fig. 5 there is a node of the keyword 'hoarse', a short passage (in fact, a sentence in this case) from J. K. Rowling's Harry Potter and the Chamber of Secrets, and a short 'Comments' section including the vocabulary entry 'shout oneself *hoarse*' which contains a hypertext link (represented by an asterisk) to the same node to which the context belongs, i.e. to the node of 'hoarse'. There are two other links in the attached context (represented by a double arrow in superscript position, just at the end of the context) which point to the bibliographic description of the sources of the context (i.e. J. K. Rowling's corresponding work and its Hungarian translation) which can be found in the Reference page.

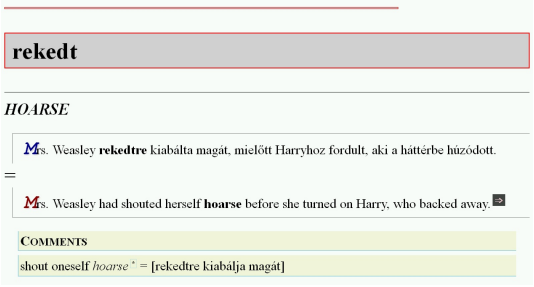


Figure 5. *The node of the keyword 'hoarse' with a bilingual context and its 'Comments' section.*

Apart from the nodes of keywords and the hypertext links in the vocabulary entries which point to them, we created specific navigation sections within the learning material each of which contains a dedicated group of hypertext links to specific parts of the material. In the following we would like to present them one by one.

First, the 'Sources' section lists a characteristic part (e.g. the first few words) of every context which occurs in the material. We grouped the items by the corresponding works of literature where the contexts occur and added several hypertext links to the items, which point to

- the bibliographic description of the corresponding literary work written (or translated) in English,
- the bibliographic description of the Hungarian translation of the corresponding literary work,
- the corresponding context (in English).

In case there are more than one context from a selected literary work, the referenced contexts are arranged according to their order of occurrence in the original work (Fig. 6).

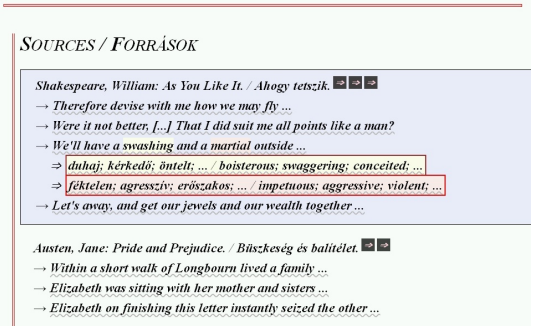


Figure 6. *The 'Sources' section containing references to the selected literature works and their contexts. Note that after the double arrows there are links to the two other navigation sections (i.e. Part 1 and Part 2, see below).*

Second, we created two other navigation sections which point to the group of keywords listed above as Part 1 and Part 2 (which are also the name of the sections themselves). The words are presented in two separate columns of a table where the second column contains the listed English words and the first one is their Hungarian equivalents. Moreover, we assigned a hypertext link to those keywords which are presented as individual nodes in the learning material (Fig. 7).

As we can see in the figure, in both columns of the table certain words are separated by horizontal lines to form subgroups of synonymous words. Where it seemed to be useful, we presented the pronunciation of some English words as well (that is, when pronouncing a word may be difficult for a Hungarian learner).

Third, we listed all the keywords and their Hungarian translations which occur in the learning material (either in a bilingual phrase or in a specific context) using a simple JavaScript program. We arranged the English keywords alphabetically and inserted a hypertext link to the exact place in the learning material where the presented keyword occurs (represented by an arrow in the third column of the table in Fig. 8). Note that we omitted from the table the occurrence of keywords in the sentence examples because their display is optional (as mentioned above).

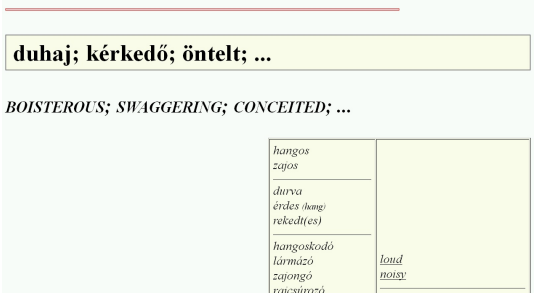


Figure 7. *The beginning of the table containing English keywords and their Hungarian equivalents from the group of words named Part 1.*

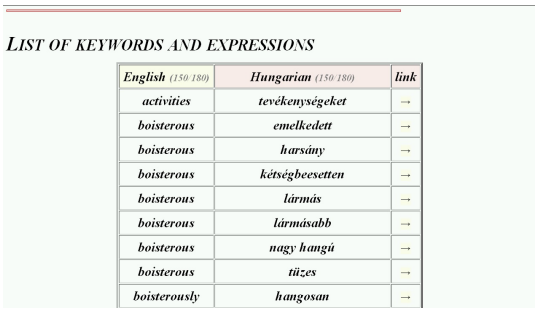
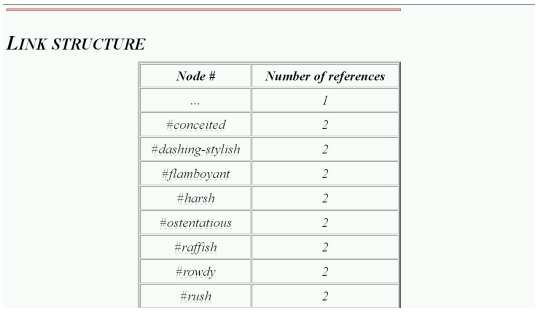


Figure 8. *The beginning of the 'List of keywords' section containing English keywords and their Hungarian translations arranged alphabetically. There are also links to the occurrence of each of them.*

Currently there are 150 occurrences of the listed keywords in the learning material (see Fig. 8).

Fourth, we listed all the English keywords which occur as separate nodes in the learning material (identified by their name after a hash mark like #conceited, #harsh etc.) using also a simple JavaScript program. We arranged the keywords by the *number of links* (called either 'Number of references' or 'Link strength') that point to the node of the respective keyword in the learning material, and inserted a hypertext link to each node represented by a gray dotted line which underlines each keyword in the first column of the table (Fig. 9). Note that we omitted those keywords the link strength of which is only 1 because of their number (actually, there are currently more than 200 such nodes).

Finally, we summarized the basic features of the *network* of nodes and hypertext links established in the bilingual language learning material. Using a JavaScript program we divided all referenced nodes of the learning material into separate groups according to the number of references which each node has (called 'Link strength') and determined the number of nodes in each group (called 'Strength



LINK STRUCTURE

Node #	Number of references
...	1
#conceited	2
#dashing-stylish	2
#flamboyant	2
#harsh	2
#ostentatious	2
#raffish	2
#rowdy	2
#rush	2

Figure 9. *The beginning of the 'Link structure' section containing the keywords (in the 'Node #' column) and the number of hypertext links that point to them (in the 'Number of references' column).*

frequency'). In the table shown in Fig. 10 we presented for each group of nodes the link strength value in the first column, and the number of nodes in the second column.

5. Evaluation and further use

In the science of networks the degree distribution of the so-called scale-free networks can be displayed by a curve that follows the power law and can therefore be described by the formula

$$N(k) = c * k^{-}\gamma \tag{5.1}$$

where N(k) is the degree or frequency of nodes that have exactly 'k' links. In other words, formula (5.1) describes the number of those nodes the "link strength" of which is exactly 'k' (see the first and second column of the table in Fig. 10). The parameters denoted by 'c' and γ are fixed parameters that characterize the specific network.

Note that the empirical value of the parameter γ (i.e. the degree exponent of the curve) for a lot of well-known scale-free networks is typically $2<\gamma<3$, e.g. $\gamma=2.5$ [1].

The JavaScript program we created fits a curve following the power law distribution of the number of nodes having exactly 'k' links described in formula (5.1) according to the series of data points presented in the first and second columns of the table in Fig. 10. The *estimated frequency values* that the fitted curve provides are presented in the third column of the table.

We found that in the current stage of the development of the learning material the value of the parameter γ is about 4 and the square root of the residual sum of squares (which, using the least squares fitting method, characterizes the deviation of the calculated values from the actual ones) is relatively high ($\Delta \approxeq 5.564$; see Fig. 10).

Link strength	*Strength frequency*	*Estimated value* (γ=4.188, Δ≈5.564)
1	*242*	*241.97*
2	*12*	*13.28*
3	*5*	*2.43*
4	*4*	*0.73*
5	*2*	*0.29*
6	*3*	*0.13*
7	*1*	*0.07*
Number of links: 332		

Figure 10. *The basic features of the network structure of the nodes of keywords and hypertext links. For example, there are 242 nodes that have 1 reference, 12 nodes that have 2 references etc.*

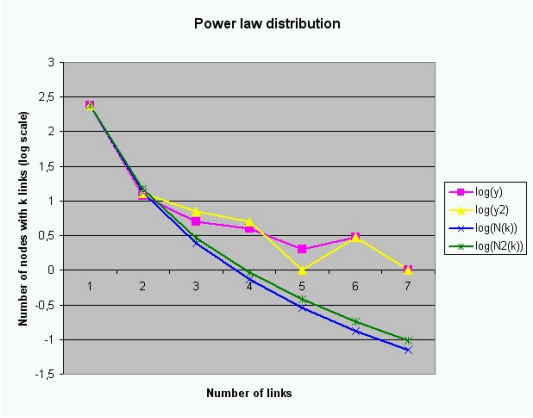


Figure 11. *The distribution of the strength frequency of nodes having exactly 'k' links on a logarithmic scale according to the data presented in Fig. 10 (first curve), and to the improved data after two weeks (second curve). The third and fourth curves have been fitted to the data in Fig. 10 and the improved data, respectively.*

However, we experienced in the content development process that during the elaboration of the language learning material the parameter γ tends to be gradually decreasing. For example, after two weeks' development of the content of the learning material, we calculated a somewhat lesser value for the parameter γ compared to the value presented in Fig. 10 (i.e. $\gamma \approxeq 4.014$ instead of $\gamma \approxeq 4.188$). So we guess that further elaboration of the material (for example inserting even more contexts

etc.) will result in an effect that the value for the exponent will be between the 'experimental' boundaries (i.e. between 2 and 3) and the deviation of the actual values from the calculated ones will be considerably less.

As for the effectiveness of the language learning material we intend to make it available freely through the internet. Both the usage statistics for a given period of time and the comments of the users can help us evaluate and improve the learning material. Note that the bilingual language learning material is also an inherent part of the 3DVLM which uses the MaxWhere Seminar System. Note that MaxWhere, on the one hand, is a desktop virtual environment for education and learning [4] which can provide, among other things, personalized, customizable learning environment and paths [22] for the learners, and, on the other hand, MaxWhere can be considered as a possible candidate for next generation 3D operation systems [24]. Besides, there are two firm pillars on which our work is founded: the 3D virtual environment might enhance the effective use of our long term memory serving as a kind of memory palace [18] and, supposing that the organization of the content elements to be memorized is more or less adequately reflected in the mental image created in the memory during the learning process, *establishing the learning material as a scale-free network of content elements* might transfer the network's high degree of robustness [1] against "memory failures" (e.g. oblivion) to the "network of knowledge" that the learners had successfully built using our learning material.

As a conclusion of those considerations we can plausibly expect that advanced (as well as enthusiastic and interested) language learners can use our learning material effectively either for self-study or in language classrooms for advanced language courses.

Acknowledgements. The results presented in this paper have partially been achieved in the Virtual Reality Laboratory of the Faculty of Informatics of the University of Debrecen, Hungary. This work has been supported by Qos-HPC-IoT Laboratory and project TKP2021 NKTA of the University of Debrecen, Hungary. Project no. TKP2021-NKTA-34 has been implemented with the support provided from the National Research, Development and Innovation Fund of Hungary, financed under the TKP2021-NKTA funding scheme."

References

[1] A.-L. Barabási: *Linked. The New Science of Networks*, Cambridge, MA: Perseus Publishing, 2002.

[2] P. Baranyi, Á. Csapó: *Definition and synergies of Cognitive Infocommunications*, Acta Polytechnica Hungarica 9.1 (2012), pp. 67–83.

[3] P. Baranyi, Á. csapó, G. Sallai: *Cognitive Infocommunications (CogInfoCom)*, Berlin, Heidelberg: Springer, 2015, doi: https://doi.org/10.1007/978-3-319-19608-4.

[4] B. Berki: *Desktop VR as a Virtual Workspace: a Cognitive Aspect*, Acta Polytechnica Hungarica 16.2 (2019), pp. 219–231.

[5] B. Berki: *Navigation Power of MaxWhere: a Unique Solution*, in: CogInfoCom 2020. Proceedings of the 11th IEEE International Conference on Cognitive Infocommunications, Piscataway, NJ, USA: IEEE, 2020, pp. 511–515, doi: https://doi.org/10.1109/CogInfoCom50765.2020.9237904.

[6] B. Berki: *Sense of Presence in MaxWhere Virtual Reality*, in: CogInfoCom 2019. Proceedings of the 10th IEEE International Conference on Cognitive Infocommunications, Piscataway, NJ, USA: IEEE, 2019, pp. 91–94, doi: https://doi.org/10.1109/CogInfoCom47531.2019.9089976.

[7] I. Boda, M. Bényei, E. Tóth: *New dimensions of an ancient Library: the Library of Alexandria*, in: CogInfoCom2013. Proceedings of the 4th IEEE International Conference on Cognitive Infocommunications, New York, NY, USA: IEEE, 2013, pp. 537–542, doi: https://doi.org/10.1109/CogInfoCom.2013.6719306.

[8] I. Boda, E. Tóth: *From Callimachus to the Wikipedia: an ancient method for the representation of knowledge in the WWW era*, in: CogInfoCom2018. Proceedings of the 9th IEEE International Conference on Cognitive Infocommunications, Piscataway, NJ, USA: IEEE, 2018, pp. 205–210, doi: https://doi.org/10.1109/CogInfoCom.2018.8639895.

[9] I. Boda, E. Tóth, M. Bényei, I. Csont: *A three-dimensional virtual library model of the ancient Library of Alexandria*, in: ICAI 2014. Proceedings of the 9th International Conference on Applied Informatics, Eger, Hungary: Eszterházy Károly Teacher Training College, 2014, vol. 1, 103–111, doi: https://doi.org/10.14794/ICAI.9.2014.1.103.

[10] I. Boda, E. Tóth, I. Csont, L. T. Nagy: *Developing a knowledge base of ancient literary texts in virtual space*, in: CogInfoCom2016. Proceedings of the 7th IEEE International Conference on Cognitive Infocommunications, Piscataway, NJ, USA: IEEE, 2016, pp. 263–270, doi: https:doi.org/10.1109/CogInfoCom.2016.7804559.

[11] I. Boda, E. Tóth, I. Csont, L. T. Nagy: *The use of mythological content in virtual learning environment*, in: CogInfoCom2017. Proceedings of the 8th IEEE International Conference on Cognitive Infocommunications, Piscataway, NJ, USA: IEEE, 2017, pp. 307–314, doi: https://doi.org/10.1109/CogInfoCom.2017.8268262.

[12] I. Boda, E. Tóth, I. Csont, L. T. Nagy: *Toward a knowledge base of literary content focusing on the ancient Library of Alexandria in the three dimensional space*, in: CogInfoCom2015. Proceedings of the 6th IEEE International Conference on Cognitive Infocommunications, New York, NY, USA: IEEE, 2015, pp. 251–258, doi: https://doi.org/10.1109/CogInfoCom.2015.7390600.

[13] I. Boda, E. Tóth, F. Z. Iszály: *Text-based approach to second language learning in the virtual space focusing on Callimachus' life and works*, in: CogInfoCom 2019. Proceedings of the 10th IEEE International Conference on Cognitive Infocommunications, Piscataway, NJ, USA: IEEE, 2019, pp. 439–444, doi: https://doi.org/10.1109/CogInfoCom47531.2019.9089933.

[14] I. K. Boda, E. Tóth: *Classical Heritage and Text-Based Second Language Learning in Three-Dimensional Virtual Library Environment*, in: ICAI 2020. Proceedings of the 11th International Conference on Applied Informatics, Achen, Germany: CEUR-WS, Vol. 2650., 2020, pp. 46–56.

[15] I. K. Boda, E. Tóth: *Content development for second language learning in the 3D virtual space*, in: CogInfoCom 2021. Proceedings of the 12th IEEE International Conference on Cognitive Infocommunications, Piscataway, NJ, USA: IEEE, 2021, pp. 321–327.

[16] I. K. Boda, E. Tóth: *English language learning by visualizing the literary content of a knowledge base in the three-dimensional space*, Annales Mathematicae et Informaticae 53 (2021), pp. 45–59, doi: https://doi.org/10.33039/ami.2021.04.003.

[17] I. K. Boda, E. Tóth: *English language learning in virtual 3D space by visualizing the library content of ancient texts*, in: CogInfoCom2020. Proceedings of the 11th IEEE International Conference on Cognitive Infocommunications, Piscataway, NJ, USA: IEEE, 2020, pp. 305–311, doi: https://doi.org/10.1109/CogInfoCom50765.2020.9237887.

[18] Á. B. Csapó, I. Horváth, P. Galambos, P. Baranyi: *VR as a Medium of Communication: from Memory Palaces to Comprehensive Memory Management*, in: CogInfoCom 2018. Proceedings of the 9th IEEE International Conference on Cognitive Infocommunications, Piscataway, NJ, USA: IEEE, 2018, pp. 389–394, DOI: https://doi.org/10.1109/CogInfoCom.2018.8639896.

[19] A. Gilányi, A. Rácz, A. M. Bólya, J. Décsei, K. Chmielewska: *A Presentation Room in the Virtual Building of the First National Theater of Hungary*, in: CogInfoCom 2020. Proceedings of the 11th IEEE International Conference on Cognitive Infocommunications, Piscataway, NJ, USA: IEEE, 2020, pp. 519–523, DOI: https://doi.org/10.1109/CogInfoCom50765.2020.9237828.

[20] I. Horváth: *An Analysis of Personalized Learning Opportunities in 3D VR*, Frontiers in Computer Science 3 (2021), pp. 1–12.

[21] I. Horváth: *How to Develop Excellent Educational Content for 3D VR*, in: CogInfoCom 2019. Proceedings of the 10th IEEE International Conference on Cognitive Infocommunications, Piscataway, NJ, USA: IEEE, 2019, pp. 483–489, DOI: https://doi.org/10.1109/CogInfoCom47531.2019.9089916.

[22] I. Horváth: *Personalized Learning Opportunity in 3D VR*, in: CogInfoCom 2020. Proceedings of the 11th IEEE International Conference on Cognitive Infocommunications, Piscataway, NJ, USA: IEEE, 2020, pp. 425–439, DOI: https://doi.org/10.1109/CogInfoCom50765.2020.9237895.

[23] *Interactive map of the three dimensional virtual library (3DVLM)*, URL: https://bodaistvan.hu/callimachus/map.html (visited on 10/30/2022).

[24] D. Kiss, P. Baranyi: *3D Webspace VS 2D Website*, in: CogInfoCom 2020. Proceedings of the 11th IEEE International Conference on Cognitive Infocommunications, Piscataway, NJ, USA: IEEE, 2020, pp. 517–518, DOI: https://doi.org/10.1109/CogInfoCom50765.2020.9237898.

[25] *Magyar-angol szó-, kifejezés- és mondattár. / English-Hungarian words, phrases and sentences. [Bilingual language learning material.]* URL: https://bodaistvan.hu/callimachus/texts/_hun-eng-01.html (visited on 11/04/2022).

[26] *MaxWhere VR Even more.*

[27] A. Rácz, A. Gilányi, A. M. Bólya, J. Décsei, K. Chmielewska: *On a Model of the First National Theater of Hungary in MaxWhere*, in: CogInfoCom 2020. Proceedings of the 11th IEEE International Conference on Cognitive Infocommunications, Piscataway, NJ, USA: IEEE, 2020, pp. 575–576, DOI: https://doi.org/10.1109/CogInfoCom50765.2020.9237848.

Annales Mathematicae et Informaticae
56 (2022) pp. 36–46
DOI: https://doi.org/10.33039/ami.2022.12.012
URL: https://ami.uni-eszterhazy.hu

Using irreducible polynomials for random number generation

Tamás Herendi[a*†], **Sándor Roland Major**[b*]

[a]Department of Computer Science
University of Debrecen
Debrecen, Hungary
herendi.tamas@inf.unideb.hu

[b]Department of Information Technology
University of Debrecen
Debrecen, Hungary
major.sandor@inf.unideb.hu

Abstract. A method is presented for generating random numbers with uniform distribution using linear recurrence sequences with very large period lengths. This method requires an irreducible polynomial modulo 2 to define the sequence. A suitable method for generating an infinite number of such polynomials is presented. The polynomials generated in this way can have an arbitrarily large degree, and a large enough order to make them suitable for practical applications.

Keywords: irreducible polynomials, finite fields, random number generation

AMS Subject Classification: 12-08 Computational methods for problems pertaining to field theory

1. Introduction

Pseudorandom number generation (PRNG) is an important component of many practical applications. Generators with different properties are used in a wide

*The author has been partially supported by the construction EFOP-3.6.3-VEKOP-16-2017-00002. The project was supported by the European Union, co-financed by the European Social Fund.

†The author has been partially supported by the SETIT Project (no. 2018-1.2.1-NKP-2018-00004), which has been implemented with the support provided by the National Research, Development and Innovation Fund of Hungary, financed under the 2018-1.2.1-NKP funding scheme.

Submitted: November 2, 2022
Accepted: December 3, 2022
Published online: December 28, 2022

range of fields, such as simulations [12], Monte Carlo methods [14]. Morover, PRNGs recently play an essential role in many areas of cryptography, for example, key generation, stream ciphers, asymmetric cryptosystems, and authentication protocols [11].

The results presented in this paper relate to an algorithm detailed in [8], showing the construction of uniformly distributed linear recurrence sequences (LRS) modulo powers of 2, with theoretically arbitrarily large period lengths. A modified version of this algorithm is given in Section 3, optimizing it to be less computationally expensive.

2. Theory

The algorithm presented takes an irreducible polynomial over $\mathbb{F}_2$ as input.

Irreducible polynomials over finite fields are used in a wide variety of contexts, not just in pure mathematics and many areas of computer science, but practical applications as well.

In [8], we see a method of constructing linear recurring sequences with extremely long periods used for pseudorandom number generation. The sequence requires an irreducible polynomial to create, the degree of which is directly related to the resulting period length.

In coding theory, creating error correcting codes that can be used to reliably transmit information over noisy channels is a key practical application that can be found in many everyday electronic systems. These codes are almost always connected to the use of polynomials over finite fields. An in-depth discussion can be found in [9].

In cryptography, many encryption protocols use finite fields as their domain. Irreducible polynomials have been used in public key cryptosystems for decades, such as in [4].

As the previous examples show, irreducible polynomials over the finite field $\mathbb{F}_2$ are of special interest.

2.1. Irreducible polynomials

A univariate polynomial over the finite field $\mathbb{F}_2$ is

$$p(x) = \sum_{k=0}^{n} a_k x^k, \;\; a_0, \ldots, a_n \in \mathbb{F}_2 \,.$$

$\mathbb{F}_2[x]$ is the set of all polynomials over $\mathbb{F}_2$. A polynomial $p \in \mathbb{F}_2[x]$ of degree k is irreducible if it has no nontrivial factors over $\mathbb{F}_2$. That is, $p(x) = p_1(x)p_2(x)$ can not hold if $\deg(p_1), \deg(p_2) > 0$. The natural way to prove a polynomial's irreducibility is, therefore, to factor it and show that no such factors can be found.

The first algorithm for factoring a polynomial over a finite field was published by Berlekamp [2]. It is a deterministic algorithm that requires a square-free polynomial, and is well suited for cases where the cardinality of the finite field is small.

Later, the Cantor-Zassenhaus algorithm [3] provided a practical solution even for polynomials over large finite fields. This algorithm is probabilistic in nature. A detailed description of both methods can be found in [13].

Rabin's test [16] provides a very simple algorithm. A polynomial over $\mathbb{F}_2$ is irreducible if and only if:

1. $p(x) \mid x^{2^k} - x$
2. $\forall t_i \ \ \mathrm{GCD}(x^{2^{k/t_i}}, p(x)) = 1,$

where t_i are the prime divisors of k. The test simply computes all $x^{2^{k/t_i}} \bmod p(x)$ polynomials using repeated squaring, and polynomial modulo operations, then uses polynomial GCD to check condition 2.

Ben-Or's test [1] modifies this approach by computing $\mathrm{GCD}(x^{2^i} \bmod p(x), p(x))$ for every $i \in \{1, \ldots, \frac{n}{2}\}$. In practice, this improves average performance when testing random polynomials. A randomly selected polynomial is much more likely to have factors of small degrees than be the product of only large-degree factors. Since Ben-Or's test checks for factors of small degrees first, these polynomials are very quickly eliminated. A comparison between the performance of Rabin's test and Ben-Or's test can be found in [7]. Victor Shoup also published a deterministic irreducibility test in [19], and a probabilistic algorithm is [18].

3. Algorithm for creating LRS

The following algorithm is for constructing uniformly distributed linear recurrence sequences modulo 2^s, with very large period lengths. It is a modified version of the algorithm found in [8]. The version presented here is significantly less computationally expensive than the original, which enables the creation of sequences with larger period lengths.

The reduced time complexity speeds up the process of finding the desired coefficients for the LRS, while the reduced space complexity allows the algorithm to be carried out with significantly larger input parameters. Once the LRS is constructed, using it to generate the pseudorandom number sequence is unchanged compared to the original version.

1. Choose an integer k and find a monic polynomial $q(x) \in \mathbb{Z}[x]$ of degree k, which reduction modulo 2 is irreducible in $\mathbb{F}_2[x]$.
2. Calculate the polynomials $p(x)$ of degree $k+2$ and $p'(x)$ of degree $k+1$ in the following way:

$$p(x) \equiv (x^2 - 1)q(x) \bmod 2 \quad \text{and}$$
$$p'(x) \equiv (x - 1)q(x) \bmod 2,$$

with the coefficients of $p(x)$ and $p'(x)$ in $\{0, -1\}$, except for the leading coefficients.

Calculate the four candidate polynomials:

$$\begin{aligned} p_1(x) &= p(x) \\ p_2(x) &= p(x) - 2 \\ p_3(x) &= p(x) - 2x \\ p_4(x) &= p(x) - 2x - 2 \end{aligned}$$

We remark that the coefficients of the constant and linear terms of the candidate polynomials can be in $\{0, -1, -2, -3\}$.

3. For $i \in \{1, 2, 3, 4\}$, $j \in \{0, 1, \dots k + 2\}$, let a_{ij} denote the coefficient of x^j in the polynomial $p_i(x)$. Calculate $S_i = \sum_{j=0}^{k+1} -a_{ij}$ for each candidate polynomial. Keep the two candidates that satisfy $S_i \equiv 1 \mod 4$. Denote these two polynomials with c_1 and c_2.

4. Let $\varrho = \mathrm{ord}(q)$ be the order of $q(x)$, i.e., the smallest positive integer such that $q(x) \mid x^\varrho - 1$.

   We need to find the candidate that satisfies $c_i(x) \nmid x^{2\varrho} - 1 \mod 4$. To do this, calculate

   $$r(x) \equiv x^\varrho \bmod (2, p(x)),$$

   where $\mathrm{mod}(2, p(x))$ means calculating the polynomial remainder with $p(x)$ over $\mathbb{F}_2$.

   Then, find the candidate that satisfies

   $$1 \not\equiv r(x)^2 \bmod (4, c_i(x)),$$

   where $\mathrm{mod}(4, c_i(x))$ means calculating the polynomial remainder with $c_i(x)$ over $\mathbb{F}_4$.

   Note that all of the computation in this step can be performed over $\mathbb{F}_2$, with the exception of the last step, which is performed over $\mathbb{F}_4$.

   Denote the candidate that remains by $c(x)$. This is the characteristic polynomial of the linear recurrence sequence we want to create. Let b_j, $j \in \{0, 1, \dots k + 2\}$ be the coefficient of x^j in $c(x)$. Then, our final recurrence relation is

   $$u_{n+k+2} = -b_{k+1}u_{n+k+1} - b_k u_{n+k} \dots - b_0 u_n$$

5. Choose initial values for the sequence. Suppose we want s-bit long pseudorandom numbers. Choose random $u_0, u_1, \dots, u_k \in [0, 2^s - 1]$. Set these values as the initial values of the linear recurrence relation with characteristic polynomial $p'(x)$. Compute the next element of the sequence, u'_{k+1}. Find a random number $u_{k+1} \in [0, 2^s - 1]$ such that $u'_{k+1} \not\equiv u_{k+1} \mod 2$.

   Set $u_0, u_1, \dots, u_{k+1}$ as the initial values of the sequence.

The original version of the above algorithm differs in steps 3 and 4. That algorithm requires computation using the companion matrices of the candidates. The companion matrix of $p_i(x)$ is

$$M_{(i)} = \begin{pmatrix} 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \\ 0 & 0 & \cdots & 0 & 1 \\ -a_{i0} & -a_{i1} & \cdots & -a_{ik} & -a_{ik+1} \end{pmatrix}$$

Step 3 calls for finding the two candidates that satisfy $M_{(i)}\bar{1} \equiv \bar{1} \bmod 4$, where $\bar{1}$ is a vector of size $k+2$ with all coordinates equal to 1. It is simple to see that this is equivalent to the step 3 of the above algorithm.

In step 4 of the old algorithm, the final candidate is the one for which $M_{(i)}^{2\varrho} \not\equiv E \pmod 4$ holds, where E is the identity matrix of size $(k+2)\times(k+2)$. This requires matrix exponentiation modulo 4, using matrices sized $(k+2)\times(k+2)$. Since we want k to be as large as possible, these matrices quickly become inconvenient, both to store, and to perform multiplications on. The step 4, presented here, instead requires mainly polynomial modulo and squaring over $\mathbb{F}_2$, and once over $\mathbb{F}_4$. This makes storage more efficient, since the size of a polynomial is a linear function of its degree. Moreover, even a naive implementation of polynomial squaring over $\mathbb{F}_2$ has time complexity $\mathcal{O}(n)$. For polynomial modulo, a naive approach has time complexity $\mathcal{O}(n^2)$, but faster algorithms are known, such as the result by Schönhage in [17], which enables polynomial division with remainder in $\mathcal{O}(n \log n \log\log n)$ time. This is better than even the fastest current matrix multiplication algorithms, such as [6], which has a complexity over $\mathcal{O}(n^{2.3})$.

4. Q-transform

A promising concept for constructing uniformly distributed high order linear recurring sequences is the application of the theory of Q-transform. In this section, we introduce some definitions and results that allow us to formulate infinite series of irreducible polynomials. Based on the idea described in Section 3, we can use such polynomials for creating uniformly distributed pseudorandom sequences with large period lengths.

During the section, q is a prime power, $\mathbb{F}$ denotes a field, $\mathbb{F}_q$ is a finite field of q elements, and $\mathbb{K}$ is an algebraic extension field of $\mathbb{F}$ or $\mathbb{F}_q$, depending on the context.

Definition 4.1. Let $p \in \mathbb{F}[x]$ be a polynomial of degree d. We say that the reciprocal polynomial of p is $p^*(x) = x^d p(x^{-1})$. We call a polynomial p self-reciprocal, if $p = p^*$.

Remark 4.2. a) If $p \in \mathbb{F}[x]$, then $p^* \in \mathbb{F}[x]$ and $(p^*)^* = p$.

b) Let $p, r, s \in \mathbb{F}[x]$ be such that $s = p \cdot r$. Then $s^* = p^* \cdot r^*$.

c) For any $p \in \mathbb{F}[x]$, p is irreducible if and only if p^* is irreducible.

d) Let $p, r, s \in \mathbb{F}[x]$ be such that $s = p \cdot r$. If p and r are self-reciprocal, then s is self-reciprocal, as well.

Proposition 1. *Let $p \in \mathbb{F}[x]$, and $\mathbb{K}$ be the splitting field of p. Then the following statements are equivalent.*

a) p is self-reciprocal;

b) $\forall \alpha \in \mathbb{K} \setminus \{0\}$: $p(\alpha) = 0$ implies $p\big(\alpha^{-1}\big) = 0$.

Corollary 4.3. *If $p \in \mathbb{F}[x]$ is self-reciprocal and irreducible of odd degree, then $p(x) = ax + a$, with some $a \in \mathbb{F}$.*

Proof. Since $\deg(p)$ is odd, Proposition 1 implies that p has a root α such that $\alpha = \alpha^{-1}$. This is possible if and only if $\alpha \in \{-1, 1\}$. Then either $x - 1$ or $x + 1$ is a divisor of p. However, p is irreducible, thus either $p = ax - a$ or $p = ax + a$, but $ax - a$ is self-reciprocal if and only if $a = -a$. □

Corollary 4.4. *Let $p \in \mathbb{F}[x]$ be a self reciprocal polynomial, and $p_1, \ldots, p_k \in \mathbb{F}[x]$ be distinct irreducible polynomials such that $p = p_1^{n_1} \cdot \cdots \cdot p_k^{n_k}$. Then for each $1 \leq i \leq k$ there exists $1 \leq j \leq k$ such that $p_i = p_j^*$ and $n_i = n_j$.*

Remark 4.5. In the previous corollary, $i = j$ if and only if p_i is self-reciprocal.

Definition 4.6. Let $p \in \mathbb{F}[x]$ be a polynomial of degree d. The Q-transform of p is $\tilde{p}(x) = x^d p(x + x^{-1})$.

Remark 4.7. If $p \in \mathbb{F}[x]$, then $\tilde{p} \in \mathbb{F}[x]$, and $\deg(\tilde{p}) = 2\deg(p)$.

Proposition 2. *Let $p, r, s \in \mathbb{F}[x]$ be such that $s = p \cdot r$. Then $\tilde{s} = \tilde{p} \cdot \tilde{r}$.*

Let $p \in \mathbb{F}[x]$, and $\alpha \in K \setminus \{0\}$. Then $\tilde{p}(\alpha) = 0$ if and only if $\tilde{p}(\alpha^{-1}) = 0$. By Proposition 1, we may state the following.

Proposition 3. *If $p \in \mathbb{F}[x]$, then $\tilde{p}$ is self-reciprocal.*

Proposition 4. *The Q-transform is an injection.*

Proof. Let $p \in \mathbb{F}[x]$, $d = \deg(p)$, $\mathbb{K}$ be the splitting field of $\tilde{p}$, and $\alpha_i, \beta_i \in \mathbb{K}$ $(i = 1, \ldots, d)$ with the following properties:

$$p(x) = a_d \prod_{i=1}^{d} (x - \alpha_i), \quad \text{and} \quad \beta_i = -\frac{1}{2}\alpha_i + \frac{1}{2}\sqrt{\alpha_i^2 - 4}\,.$$

Then

$$\tilde{p}(x) = a_d x^d \prod_{i=1}^{d} \big(x + x^{-1} - \alpha_i\big) = a_d \prod_{i=1}^{d} \big(x^2 + 1 - \alpha_i x\big)$$

$$= a_d \prod_{i=1}^{d} (x - \beta_i) \prod_{i=1}^{d} \left(x - \beta_i^{-1}\right).$$

This means that there is a one-to-one correspondence between the roots of p and the pairs of roots of $\tilde{p}$. □

Let

$$\mathcal{P}_q(d) = \{p \mid p \in \mathbb{F}_q[x], \deg(p) = d\},$$
$$\mathcal{Q}_q(d) = \{p \mid p \in \mathbb{F}_q[x], \deg(p) = 2d, p = p^*\}.$$

Since $|\mathcal{P}_q(d)| = |\mathcal{Q}_q(d)|$, Proposition 4 implies the following.

Corollary 4.8. *Let $p \in \mathbb{F}_q[x]$ be a self-reciprocal polynomial. Then there exists a unique $r \in \mathbb{F}_q[x]$ such that $p = \tilde{r}$.*

Notation 1. *Let $p \in \mathbb{F}[x]$ and $k \in \mathbb{N}$. We denote by $\tilde{p}^{(k)}$ the following iterated Q-transform:*

$$\text{if } k = 0, \text{ then } \tilde{p}^{(k)} = p;$$
$$\text{if } k > 0, \text{ then } \tilde{p}^{(k)} = \tilde{r}, \text{ where } r = \tilde{p}^{(k-1)}.$$

Corollary 4.9. *Let $p \in \mathbb{F}_q[x]$ be a self-reciprocal polynomial. Then there exists a unique $r \in \mathbb{F}_q[x]$, not a self-reciprocal polynomial, and $k \in \mathbb{N}$ such that $p = \tilde{r}^{(k)}$.*

Corollary 4.10. *Let $p \in \mathbb{F}_q[x]$ be irreducible. Then $\tilde{p}$ is either irreducible or there exist $p_1, p_2 \in \mathbb{F}_q[x]$ irreducible polynomials such that $\tilde{p} = p_1 \cdot p_2$, and $p_1 = p_2^*$.*

Proof. Assume contrary that there exists an $r \in \mathbb{F}_q[x]$ self-reciprocal polynomial with $1 \leq \deg(r) < 2\deg(p)$, such that $r|\tilde{p}$. By Corollary 4.8, there exists $s \in \mathbb{F}_q[x]$ satisfying $s \mid p$, $\deg(s) < \deg(p)$, and $r = \tilde{s}$, which is a contradiction. □

Proposition 5. *Let $p \in \mathbb{F}_2[x]$ be an irreducible polynomial in the form $p(x) = x^d + a_{d-1}x^{d-1} + \cdots + a_1 x + 1$. Then $\tilde{p}$ is irreducible if and only if $a_{d-1} = a_1 = 1$. Furthermore, the coefficient of the linear term of $\tilde{p}$ is 1.*

Proof. The proposition is proven in a more general settings in [10]. □

Corollary 4.11. *Let $p \in \mathbb{F}_2[x]$ be an irreducible polynomial, and $p(x) = x^d + x^{d-1} + a_{d-2}x^{d-2} + \cdots + a_2 x^2 + x + 1$. Then $\tilde{p}^{(k)}$ is irreducible for all $k \in \mathbb{N}$.*

This result implies that any irreducible polynomial in the form as in Proposition 5 determines an infinite sequence of irreducible Q-iterated polynomials. Every self-reciprocal polynomial of even degree is contained in exactly one of such sequences.

Proposition 6. *Let $p \in \mathbb{F}_q$ be an irreducible polynomial, accomplishing $\deg(p) = 2d$. Then p is self-reciprocal if and only if $\operatorname{ord}(p) \mid q^d + 1$.*

Proof. The proposition is stated e.g. in [5]. □

For the construction of pseudorandom number sequences with high period length, we need irreducible polynomials of high order. Actually, the period length is proportional to the order. Based on our experience, we have the following conjecture.

Conjecture 1. *Let $p \in \mathbb{F}_q[x]$ be an irreducible self-reciprocal polynomial of degree* $\deg(p) = 4d$. *Then* $q^d + 1 < \mathrm{ord}(p)$.

Furthermore, we have encountered Q-iterated polynomials having maximal order in many cases.

5. Statistical testing

In this section, we describe a test carried out to examine the statistical properties of the pseudorandom number sequences generated using the previously detailed method. Two irreducible polynomials of large degree were created, one using a brute force method and one using Q-transformations. The pseudorandom sequences generated using these polynomials were tested using the NIST statistical test suite.

The software and documentation of the NIST test suite are available at [15]. The suite includes 15 tests designed to examine the properties of pseudorandom bit sequences, such as:

- *Frequency test*: a simple check to determine the proportion of ones and zeroes in a binary sequence.
- *Runs test*: checking the number of runs (uninterrupted sequence of identical bits) of various lengths to see how closely matches the expected value in a truly random sequence.
- *DFT (Spectral) test*: determining the peak heights in the Discrete Fourier Transform of the sequence, with the purpose of finding periodic features.
- *Template matching test*: finding occurences of predetermined target strings, to detect generators producing too many such patterns. Both overlapping and non-overlapping tests are included.
- *Maurer's "Universal Statistical" test*: checking whether or not the sequence can be significantly compressed without loss of information.
- *Linear complexity test*: attempting to determine the length of the LRS that characterizes the sequence.

The first irreducible polynomial tested, denoted by t_1, was generated using irreducibility testing methods described in previous sections. The implementation uses the NTL (Number Theory Library) available at [20].

The degree of t_1 was chosen to be 216091. The reason for this choice is that $2^{216091}-1$ is a Mersenne prime. Choosing a value this way simplifies Step 4 of the algorithm described in Section 2. Note that this step requires the computation of the order of the irreducible polynomial, which is a divisor of 2^d-1, where d is the degree of the polynomial. If d is large, this step becomes computationally impractical, but choosing 2^d-1 to be a prime gives a simple solution to the problem.

The second irreducible polynomial tested, denoted t_2, was created using iterated Q-transform, using the following method:

1. Let q be a self-reciprocal irreducible monic polynomial, with $\deg(q) = d$.
2. Run the algorithm described in Section 3, using q as input. Let p be the candidate polynomial that remains after Step 4. Determine $s, r \in \mathbb{Z}[x]$ such that $p = sq + r$, and $\deg(r) < \deg(q)$.
3. Compute $t = s\tilde{q}^{(n)} + r$, where $\tilde{q}^{(n)}$ is the iterated Q-transform, described in Notation 1 of Section 4. Use t to construct the linear recurrence sequence.

Based on practical observation, if the sequence produced by p has uniform distribution, then the sequence produced by t will also have uniform distribution. However, the proof of this conjecture is currently an open question.

To create t_2, the following polynomial was used as a starting point:

$$q_2 = x^{14} + x^{13} + x^{12} + x^{11} + x^{10} + x^9 + x^7 + x^5 + x^4 + x^3 + x^2 + x + 1\,.$$

As it is stated in Proposition 6, the order of a self-reciprocal irreducible monic polynomial of degree d is at most $2^{\frac{d}{2}}+1$. The above polynomial was chosen because $\mathrm{ord}(q), \mathrm{ord}(\tilde{q})$, and $\mathrm{ord}(\tilde{q}^{(2)})$ all reach this maximum value. Theoretically, it does not guarantee that this maximality property will hold after further Q-transformations, but practical observations suggest that the order of $\tilde{q}^{(n)}$ will grow at a rate that is sufficient for use in the applications described in this paper. We stated our related experience in Conjecture 1.

Using the above method, $p_2 = s_2q_2 + r_2$ was determined, and the polynomial to be used was set as $t_2 = s_2\tilde{q_2}^{(14)} + r_2$. Note that $\deg(\tilde{q_2}^{(14)}) = 229376$, and $\deg(t_2) = 229378$.

Using t_1 and t_2, two LRSs were created to generate the pseudorandom sequences to be tested, denoted L_1 and L_2 respectively. Both LRSs generate 64-bit words. Following the recommendations in the documentation of the NIST test suite, 16MB (2^{21} words) of test data were generated using L_1 and L_2 each.

For each of these two streams, the NIST suite split the data into 100 bitstreams. The testing software provides a detailed output of the tests, as well as a summary showing the number of bitstreams that passed each test. The minimum pass rate for a test is considered to be 96 out of a sample size of 100.

Tables 1 and 2 show some of the result obtained from the tests. The full report can be found at https://arato.inf.unideb.hu/major.sandor/statistical_results/.

Table 1. NIST test results of L_1 generator.

Statistical Test	P-value	Proportion
Frequency	0.779188	100/100
Runs	0.514124	100/100
FFT	0.924076	99/100
OverlappingTemplate	0.012650	96/100
Universal	0.935716	97/100
LinearComplexity	0.699313	99/100

Table 2. NIST test results of L_2 generator.

Statistical Test	P-value	Proportion
Frequency	0.955835	100/100
Runs	0.108791	98/100
FFT	0.678686	98/100
OverlappingTemplate	0.035174	97/100
Universal	0.249284	100/100
LinearComplexity	0.719747	100/100

The results show the two generators to have very similar statistical properties, with L_2 being only slightly weaker in some tests. Since the order of t_2 is significantly lower than the order of t_1, this result is to be expected. Also of note is that both generators produced very high quality pseudorandom sequences, passing all relevant benchmarks set by the test suite.

This shows that using the Q-transformation described above to generate irreducible polynomials of very large degree is completely suitable for use in generating uniformly distributed pseudorandom linear recurrence sequences.

References

[1] M. Ben-Or: *Probabilistic Algorithms in Finite Fields*, 22nd Annual Symposium on Foundations of Computer Science (1981), pp. 394–398.

[2] E. R. Berlekamp: *Factoring Polynomials over Finite Fields*, The Bell System Technical Journal 46(8) (1967), pp. 1853–1859.

[3] D. Cantor, H. Zassenhaus: *A New Algorithm for Factoring Polynomials Over Finite Fields*, Mathematics of Computation 36(154) (1981), pp. 587–592.

[4] B. Chor, R. L. Rivest: *A Knapsack-type Public Key Cryptosystem Based on Arithmetic in Finite Fields*, IEEE Transactions on Information Theory 34(5) (1988), pp. 901–909.

[5] S. Cohen: *Polynomials over finite fields with large order and level*, Bull. Korean Math. Soc. 24(2) (1987), pp. 83–96.

[6] F. L. Gall: *Powers of tensors and fast matrix multiplication*, Proceedings of the 39th International Symposium on Symbolic and Algebraic Computation (ISSAC '14) (2014), pp. 296–303.

[7] S. Gao, D. Panario: *Tests and Constructions of Irreducible Polynomials over Finite Fields*, Foundations of Computational Mathematics (1997).

[8] T. Herendi: *Construction of Uniformly Distributed Linear Recurring Sequences Modulo Power of 2*, Uniform Distribution Theory 13(1) (2018), pp. 109–129.

[9] R. Lidl, H. Niederreiter: *Introduction to Finite Fields and their Applications*, Cambridge: Cambridge University Press, 1994.

[10] H. Mein: *On the Construction of Irreducible Self-Reciprocal Polynomials Over Finite Fields*, Communication and Computing 1 (1990), pp. 43–53.

[11] A. J. Menezes, S. A. Vanstone, P. C. V. Oorschot: *Handbook of Applied Cryptography (1st. ed.)* USA: CRC Press, Inc., 1996.

[12] S. L. Miller, D. Childers: *Probability and Random Processes (2nd. ed.)* Boston: Academic Press, 2012, pp. 517–546.

[13] P. Naudin, Q. Claude: *Univariate polynomial factorization over finite fields*, Theoretical Computer Science 191 (1998), pp. 1–36.

[14] H. Niederreiter, Philadelphia, Pennsylvania: Society for Industrial and Applied Mathematics, 1992, pp. 161–176.

[15] NIST: *A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications*, URL: https://csrc.nist.gov/Projects/Random-Bit-Generation/Documentation-and-Software.

[16] M. Rabin: *Probabilistic Algorithms in Finite Fields*, SIAM J. Comput. 9 (1980), pp. 273–280.

[17] A. Schönhage: *Asymptotically fast algorithms for the numerical multiplication and division of polynomials with complex coefficients* (1982).

[18] V. Shoup: *Fast Construction of Irreducible Polynomials over Finite Fields*, Journal of Symbolic Computation 17(5) (1994), pp. 371–391.

[19] V. Shoup: *New Algorithm for Finding Irreducible Polynomials over Finite Fields*, Mathematics of Computation 54(189) (1990), pp. 435–447.

[20] V. Shoup: *NTL: A Library for doing Number Theory*, URL: https://libntl.org/.

Annales Mathematicae et Informaticae
56 (2022) pp. 47–57
DOI: https://doi.org/10.33039/ami.2022.12.013
URL: https://ami.uni-eszterhazy.hu

Unbounding discrete oriented polytopes

Mátyás Kiglics*, Gábor Valasek, Csaba Bálint*

Faculty of Informatics, Eötvös Loránd University, Budapest, Hungary
kiglics@caesar.elte.hu
valasek@inf.elte.hu
csabix@inf.elte.hu

Abstract. We propose an efficient algorithm to compute k-sided unbounding discrete oriented polytopes (k-UDOPs) in arbitrary dimensions. These convex polytopes are constructed for a fixed set of directions and a given center point. The interior of k-UDOPs does not intersect the scene geometry. We discuss several types of general geometric queries on these constructs, such as intersection with rays, and provide an empirical investigation on the limit of these shapes as the number of sides increases. In the 2D case, we extend our construction to planar shapes enclosed by arbitrary parametric boundaries with known derivative bounds.

Keywords: computer graphics, computational geometry, collision avoidance

AMS Subject Classification: 68U05

1. Introduction

Bounding volumes are ubiquitous in various computing venues, such as computer graphics [6], collision detection [2, 3, 7, 9], and geometric information systems.

A $B \subset \mathbb{R}^D$ volume is a bounding volume of a $G \subset \mathbb{R}^D$ geometry if $G \subset B$ holds. This property facilitates quick filtering of geometries, in other words, we only execute a query on G if it is successful on B. For example, if a line does not intersect B, it cannot intersect G.

The more efficiently a query is carried out on B, the more performance may be gained by using culling based on bounding volumes. However, B has to be a

*EFOP-3.6.3-VEKOP-16-2017-00001: Talent Management in Autonomous Vehicle Control Technologies – The Project is supported by the Hungarian Government and co-financed by the European Social Fund. Supported by the ÚNKP-21-3 New National Excellence Program of the Ministry for Innovation and Technology from the source of the National Research, Development and Innovation Fund.

Submitted: November 6, 2022
Accepted: December 3, 2022
Published online: December 28, 2022

sufficiently close approximation to the shape of G to avoid an excessive amount of false positives. In practice, bounding volumes are also organized into hierarchies [6], a construct that only depends on the structure of the initial bounding volumes, not the geometries they contain.

Typical realization of bounding volumes are axis-aligned bounding boxes (AABBs) and oriented bounding boxes (OBBs). Geometric queries, for example, ray-surface intersection and collision detection against other similar volumes, are trivially resolved on these at the expense of their relatively low capacity for adapting to the shape and orientation of their enclosed geometries. These properties are improved by generalizing bounding boxes to the k-sided bounding discrete oriented polytope (k-DOP). The k-DOPs are defined as the intersection of k half-spaces; as such, they are convex. As k increases, the bounding volume can better adapt to the source geometry. However, it also incurs additional processing costs upon filtering geometries, as we must process more faces. In this sense, the choice of k is a trade-off between adaptivity and query complexity.

Unbounding volumes are complements to bounding volumes. They enclose empty spaces such that none of their interior points intersect any geometry. The most prominent example of such a construct is Hart's sphere tracing [5] algorithm that infers unbounding spheres from signed distance values to accelerate ray tracing.

In collision detection and path planning, these unbounding volumes can be used to reject geometries that cannot intersect a given entity. In this case, an unbounding geometry with a smaller volume generates fewer candidates in the filtering pass; thus, it also decreases the number of false positives.

Section 2 describes an efficient algorithm to compute k-sided unbounding convex oriented polytopes, or k-UDOPs, for a wide range of geometry types. The construction runs in $\Theta(kN)$ complexity for N objects and is generalized to higher dimensions. Our algorithm relies on the capability to compute the distance of the discussed geometries to hyperplanes. Section 3 enumerates several simple geometric representations and how to compute these distances on them. In particular, Section 3.5 describes a method to infer a conservative k-UDOP for shapes with arbitrary parametric boundaries in the plane, given known bounds on their derivatives, and in Section 4, we describe some alogorithms applied on k-UDOPs. In Section 5, we demonstrate that the k-UDOP converges to a polygon empirically as the number of sides increases. Finally, we demonstrate the results of our proposed algorithm on various plane geometries in Section 5.

2. Unbounding k-DOP construction

2.1. Representation

A k-DOP, or a discrete oriented polytope with k sides in $2 \leq D \in \mathbb{N}$ dimensions, is defined by a center point $\boldsymbol{c} \in \mathbb{R}^D$, and a sequence of directions $\boldsymbol{v}_i \in \mathbb{R}^D$, $\|\boldsymbol{v}_i\|_2 = 1$ and distances $0 \leq h_i \in \mathbb{R}$, where $i = 1, \ldots, k$ and $k \geq D + 1$. The interior of the

k-DOP is given by

$$H_{\boldsymbol{c}} = \{\boldsymbol{x} \in \mathbb{R}^D \mid \forall i \in \{1, \ldots, k\} : (\boldsymbol{x} - \boldsymbol{c})^{\mathrm{T}} \cdot \boldsymbol{v}_i < h_i\}$$

For even k values, one may arrange the directions such that $\boldsymbol{v}_{2i} = -\boldsymbol{v}_{2i+1}$. This makes k-DOP queries more efficient, essentially halving the number of necessary evaluations. Note that sometimes the literature uses this convention, that is, a k-DOP refers to a $2k$ sided oriented convex polytope. In our constructs, k denotes the number of sides.

In this paper, we consider k and the $\boldsymbol{v}_i$ directions fixed and investigate the problem of finding the largest *unbounding* k-DOP around a point $\boldsymbol{c}$ that does not contain any point from a predefined set of geometries $A \subset \mathbb{R}^D$, that is $A \cap H_{\boldsymbol{c}} = \emptyset$. Note that $H_{\boldsymbol{c}}$ is convex, making intersection tests highly efficient. For example, Algorithm 2 is an $\Theta(k)$ algorithm for computing intersection with a ray.

2.2. Algorithm

We present a generalized method for constructing k-UDOPs for a fixed center point and directions. Let us consider a two-dimensional scene S, consisting of arbitrary geometric entities with a known evaluation of the distance-to-hyperplane query. Our method (Algorithm 1) is summarized as follows.

Algorithm 1 Constructing general k-UDOPs

Input: $\mathbf{c}$ center point, S set of geometries, $\mathbf{v}_i$ directions
Output: h_i distances
$h_i \leftarrow \infty$ ▷ $1 \leq i \leq k$
for all $g \in S$ **do**
  for all $\mathbf{v}_i$ **do**
    $d_i \leftarrow \min\{(\mathbf{p} - \mathbf{c})^{\mathrm{T}} \cdot \mathbf{v}_i \mid \mathbf{p} \in g\}$ ▷ distance of g from line containing $\mathbf{c}$
  end for
  if $h_i > d_i \quad \forall i \in \{j \mid d_j > 0\}$ **then** ▷ g is inside
    $m \leftarrow$ index of $\max(d_1, d_2, \ldots, d_k)$
    $h_m \leftarrow d_m$
  end if
end for

For every shape $g \in S$, we need to calculate the signed d_i distances, that is, the dot product of the $\boldsymbol{v}_i$ directions and $\boldsymbol{p} - \boldsymbol{c}$ vectors, where $\boldsymbol{p}$ is the point of g with the smallest Euclidean distance from the hyperplane defined by $\boldsymbol{c}$ and $\boldsymbol{v}_i$. The calculations of these distances are detailed in Section 3. Using these distances, we can separate the directions for which g overlaps the k-UDOP, and if that is the case for at least one $\boldsymbol{v}_i$, we overwrite h_m with the largest distance d_m. After iterating through every g shape, the $h_1, h_2, \ldots, h_k$ distances represent a single k-UDOP that does not overlap with any g. The construction method is visualized in Fig. 1, and some example scenes are presented in Fig. 2 and Fig. 3.

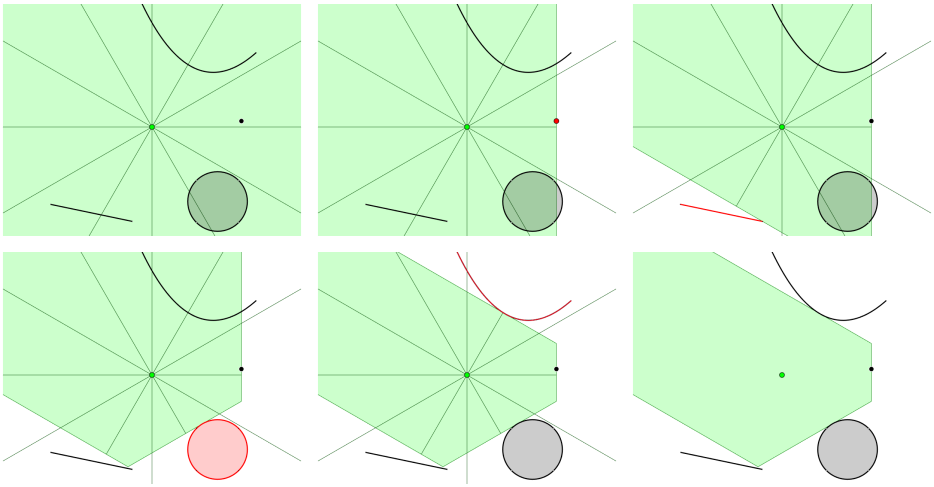


Figure 1. Construction of an unbunding 12-DOP polytope about a point (in green) to a scene containing a point, a line segment, a circle, and a quadratic Bézier curve. All h_i distances along $\boldsymbol{v}_i$ directions (thin lines) are initialized to $+\infty$ (top-left). Then at each iteration, we select a geometry (in red) and adjust the distances along all directions that have a positive dot product with the vector from the center point to the closest point of the selected geometry. The final k-UDOP have less sides than the number of directions it have started with (bottom-right).

This algorithm is linear in the number of entities for a fixed k, so its complexity is $\Theta(Nk)$, where $|S| = N$. The algorithm can be applied in higher dimensions, assuming we can evaluate the necessary signed distances.

3. Distance computations

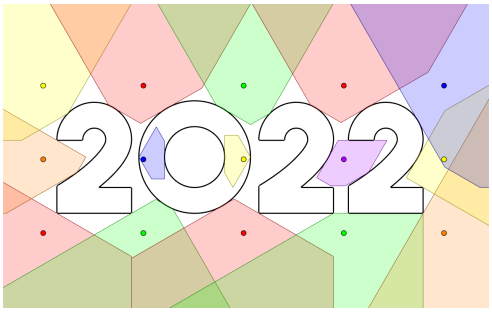
This section summarizes signed distance computations between a hyperplane and elementary geometric shapes.

For a fixed $\boldsymbol{x}_0$ and $\boldsymbol{v}_i$ direction, let $d(\boldsymbol{x})$ denote the signed distance between $\boldsymbol{x}$ and the hyperplane passing through $\boldsymbol{x}_0$ with normal $\boldsymbol{v}_i$, that is, $d(\boldsymbol{x}) = (\boldsymbol{x}-\boldsymbol{x}_0)^{\mathrm{T}} \cdot \boldsymbol{v}_i$, $\|\boldsymbol{v}_i\|_2 = 1$.

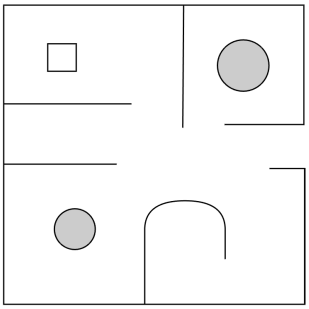
Let there be given an $\boldsymbol{x}_0$ region center and a unit normal $\boldsymbol{v}_j$ and let L_j denote the hyperplane that passes through $\boldsymbol{x}_0$ with normal $\boldsymbol{v}_j$, that is, $L_j = \{\boldsymbol{x} \in \mathbb{R}^D \mid (\boldsymbol{x} - \boldsymbol{x}_0)^{\mathrm{T}} \cdot \boldsymbol{v}_j\}$.

3.1. Points

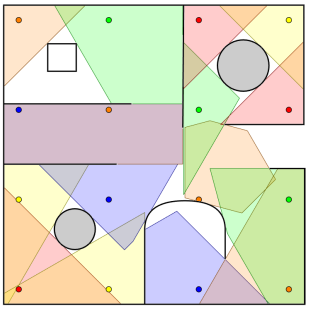
The distance from a point $\boldsymbol{x}$ is computed as $(\boldsymbol{x} - \boldsymbol{x}_0)^{\mathrm{T}} \cdot \boldsymbol{v}_j$.



(a) Unbounding 8-DOPs fitted to a text with a TrueType font. The boundary curves are composed of linear and quadratic polynomial segments.



(b) A stylized floor-plan composed of line segments, quadratic Bézier curves and circles.



(c) Fitting unbounding 12-DOPs to the floorplan of Fig. 2b.

Figure 2. Test scenes used in our deterministic tests. Figures 2a and 2c illustrate a sparse 3×5 and 4×4 grid of center points and the corresponding 15 and 16 k-UDOPs of the respective scenes.

3.2. Line segments and polygons

A line segment between $\boldsymbol{a}, \boldsymbol{b} \in \mathbb{R}^D$ is parametrized as $\boldsymbol{p}(t) = \boldsymbol{a} + t(\boldsymbol{b} - \boldsymbol{a})$, $t \in [0, 1]$. The smallest distance between L_j and $\boldsymbol{p}(t)$ is then either at their intersection point at $t = -\frac{(\boldsymbol{a}-\boldsymbol{x}_0)^{\mathrm{T}} \cdot \boldsymbol{v}_j}{(\boldsymbol{b}-\boldsymbol{a})^{\mathrm{T}} \cdot \boldsymbol{v}_j}$, if $\boldsymbol{a} \neq \boldsymbol{b}$ and $t \in [0, 1]$, or the smallest of $|d(\boldsymbol{a})|$ and $|d(\boldsymbol{b})|$.

The distance of an n-sided polygon is resolved by taking the smallest distance between L_j and the polygon edges.

3.3. Bézier curves

Let $\boldsymbol{b}(t) = \sum_{i=0}^{n} \boldsymbol{b}_i B_i^n(t)$, $t \in [0, 1]$ denote a degree n Bézier curve, where $\boldsymbol{b}_i \in \mathbb{R}^D$, $i = 1, \ldots, n$ are control points and $B_i^n(t) = \binom{n}{i} t^i (1-t)^{n-i}$ are the Bernstein polynomials.

The smallest distance is either realized at a $t^* \in [0, 1]$ parameter or at one of

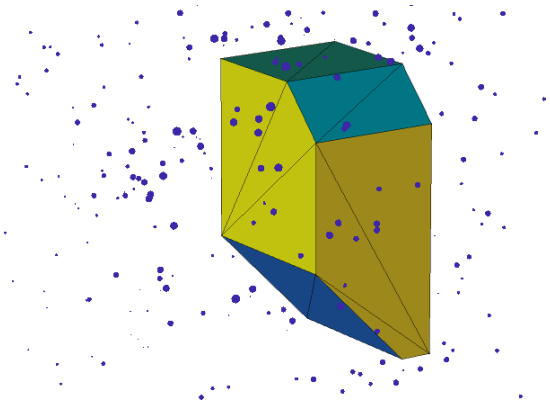


Figure 3. A 3D generalization of Algorithm 1 to a point cloud.

the curve endpoints $\boldsymbol{b}_0$ or $\boldsymbol{b}_n$. Since

$$\begin{aligned} d(\boldsymbol{b}(t)) &= (\boldsymbol{b}(t) - \boldsymbol{x})^{\mathrm{T}} \cdot \boldsymbol{v}_j \\ &= \left(\sum_{i=0}^{n} B_i^n(t) \boldsymbol{b}_i - \sum_{i=0}^{n} B_i^n(t) \boldsymbol{x} \right)^{\mathrm{T}} \cdot \boldsymbol{v}_j \\ &= \sum_{i=0}^{n} B_i^n(t) \underbrace{(\boldsymbol{b}_i - \boldsymbol{x})^{\mathrm{T}} \cdot \boldsymbol{v}_j}_{d_i} = \sum_{i=0}^{n} B_i^n(t) d_i \ , \end{aligned}$$

the parameters of the closest points on the curve satisfy

$$\partial_t d(\boldsymbol{b}(t)) = n \sum_{i=0}^{n-1} B_i^{n-1}(t) \Delta d_i = 0 \ ,$$

using the notation $\Delta^k d_i = \Delta^{k-1} d_{i+1} - \Delta^{k-1} d_i$, $k \geq 1$, $i = 0, \ldots, n-k$ and the convention $\Delta^0 d_i = d_i$.

In case of quadratic Bézier curves, the solution is

$$t = -\frac{\Delta d_0}{\Delta^2 d_0} \ ,$$

as long as $\Delta^2 d_0 \neq 0$ and t is in $[0, 1]$ interval. The closest distance is then realized either at $\boldsymbol{b}_0$, $\boldsymbol{b}_2$, or $\boldsymbol{b}(t)$ between the line and the Bézier curve.

For cubic Bézier curves, we can use the Bernstein form of the quadratic formula to obtain the two roots as

$$t_{1,2} = \frac{-\Delta d_0 \pm \sqrt{d_1^2 - d_0 d_2}}{\Delta^2 d_0} \ .$$

The minimum distance is then at either one of the roots that lie in $[0, 1]$ or at one of the endpoints.

Note that the convex hull property of Bézier curves [4] allows us to approximate the exact distance to $\boldsymbol{b}(t)$.

3.4. Spheres

The signed distance between L_j and a sphere with center $\boldsymbol{c} \in \mathbb{R}^D$ and radius $r > 0$ is $(\boldsymbol{c} - \boldsymbol{x})^{\mathrm{T}} \cdot \boldsymbol{v}_j - r$.

3.5. Shapes with continuous parametric boundaries

Let us consider the plane only and address the case of shapes that have sufficiently many times continuously differentiable boundaries, parametrized by some $\boldsymbol{p}(t) : [a, b] \to \mathbb{R}^2$ mapping. We devise conservative bounds on the distance between the line L_j and $\boldsymbol{p}(t)$ given a bound on the magnitude of the appropriate derivatives of $\boldsymbol{p}(t)$.

First, we construct a piecewise polynomial approximation to the boundary to achieve this. Afterward, we compute the distance of L_j to these polynomial boundary approximations, as shown in Section 3.3. Finally, using the error term of the particular approximating polynomial, we decrease the computed distance.

Geometrically, the last step uses a distance lower bound to the offset of the polynomial approximation. The key insight is that we do not explicitly represent the offset of the parametric boundary; it is sufficient to apply it in distance space [1].

Let us consider the case of order k Hermite interpolation. Let $\boldsymbol{h}_k(t)$ denote the Hermite polynomial that interpolates $\boldsymbol{p}^{(l)}(t_k)$, $l = 0, \ldots, k$ at prescribed knots t_k, where $\boldsymbol{p}^{(l)}(t)$ denotes the l-th derivative at t. Then

$$\boldsymbol{p}(t) - \boldsymbol{h}_k(t) = \frac{\boldsymbol{p}^{(k+1)}(\xi)}{(k+1)!}(t - t_k)^{k+1}(t_{k+1} - t)^{k+1}$$

holds for some $\xi \in (t_k, t_{k+1})$. If $M > 0$ is a bound on $\|\boldsymbol{p}^{(k+1)}\|_\infty$, then the right hand side of

$$\|\boldsymbol{p} - \boldsymbol{h}_k(t)\|_\infty \leq \underbrace{\frac{M}{(k+1)!}\left|\frac{t_{k+1} - t_k}{2}\right|^{2k}}_{E_k}$$

provides the maximum deviation between the polynomial approximation and the original shape. Computing the Hermite interpolation in Bernstein basis is trivial [4], and subtracting $\sqrt{2} \cdot E_k$ from the distance between L_j and the polynomial approximation gives a conservative bound on the distance between L_j and the segment of $\boldsymbol{p}(t)$ between t_k and t_{k+1}.

4. Queries on *k*-UDOPs

4.1. Converting *k*-UDOPs to polytope mesh

To compute the vertices of the D-dimensional polytope, we have to find the intersection of all possible combinations of D planes. This produces $\binom{k}{D}$ vertices that could be part of the polytope, so we have to filter out those that are not.

Given two vectors $\boldsymbol{a}, \boldsymbol{b} \in \mathbb{R}^D$, we can compute the $\boldsymbol{v} = \alpha \cdot \boldsymbol{a} + \beta \cdot \boldsymbol{b} \in \mathbb{R}^D$ vector that has perpendicular difference vectors to $\boldsymbol{a}$ and $\boldsymbol{b}$, with the following deltoid [8] formula:

$$\begin{bmatrix}\alpha\\ \beta\end{bmatrix} = \frac{1}{\boldsymbol{a}^{\mathrm{T}}\boldsymbol{a} \cdot \boldsymbol{b}^{\mathrm{T}}\boldsymbol{b} - \boldsymbol{a}^{\mathrm{T}}\boldsymbol{b} \cdot \boldsymbol{a}^{\mathrm{T}}\boldsymbol{b}} \begin{bmatrix}\boldsymbol{b}^{\mathrm{T}}\boldsymbol{b} \cdot (\boldsymbol{a}^{\mathrm{T}}\boldsymbol{a} - \boldsymbol{a}^{\mathrm{T}}\boldsymbol{b})\\ \boldsymbol{a}^{\mathrm{T}}\boldsymbol{a} \cdot (\boldsymbol{b}^{\mathrm{T}}\boldsymbol{b} - \boldsymbol{a}^{\mathrm{T}}\boldsymbol{b})\end{bmatrix}, \tag{4.1}$$

where the vertex is obtained with $\boldsymbol{v} = \alpha \cdot \boldsymbol{a} + \beta \cdot \boldsymbol{b}$.

Equation (4.1) allows the computation of multiple intersections in parallel. Applying the formula $D-1$ times to D directions yields a single vertex, which is then tested against the boundary of the k-DOP. Thus, in general, the algorithm is slow $O(k^{D+1})$. In two dimensions, these steps can be simplified to be $O(k^2)$ because directions have a circular ordering.

Once the vertices are computed, the connectivity information of the vertices may be computed by running a D-dimensional convex hull algorithm.

Algorithm 2 Ray and k-DOP intersection

Input: $\mathbf{c}$ center, $\mathbf{v}_i$ directions, h_i distances, $\boldsymbol{p}_0 + t\boldsymbol{d}$ ray
Output: t_1, t_2 intersection parameters
$t_1 \leftarrow -\infty$, $t_2 \leftarrow +\infty$
for all $\mathbf{v}_i$ **do**
  $t \leftarrow \dfrac{(\boldsymbol{p}_0 - \boldsymbol{c})^{\mathrm{T}} \cdot \boldsymbol{v}_i}{\boldsymbol{d}^{\mathrm{T}}\boldsymbol{v}_i}$; ▷ Intersect with each plane
  if $\boldsymbol{d}^{\mathrm{T}}\boldsymbol{v}_i < 0$ **then** ▷ Is plane back-facing?
    $t_1 \leftarrow \max(t_1, t)$ ▷ Keep furthest back-facing
  else
    $t_2 \leftarrow \min(t_2, t)$ ▷ Keep closest front-facing
  end if
end for
If $t_1 < t_2$ then there is an intersection

4.2. Ray intersection

Intersecting a k-UDOP with a ray in two dimensions can be reformulated as a ray-convex polygon intersection problem once the k-UDOP is converted to a polygon, as shown in Section 4.1. Even though the subsequent intersection computation may be carried out in $\mathcal{O}(\log k)$ time, it does not generalize to higher dimensions

and involves a quadratic time conversion. Pre-computing the polygons mitigates the latter; however, it may double the storage for each k-UDOP.

Instead, a linear time ray intersection algorithm may be formulated directly on our k-UDOP representation, shown in Algorithm 2.

The main idea is to divide the half planes into two groups: front-facing ($\boldsymbol{d}^{\mathrm{T}}\boldsymbol{v}_i > 0$), and back-facing ($\boldsymbol{d}^{\mathrm{T}}\boldsymbol{v}_i < 0$). We have to find the smallest t solution amongst the front-facing (t_2) and the largest t solution for the back-facing (t_1) planes. If, and only if, the k-DOP is intersected, then $t_1 < t_2$ and hence for any $t \in [t_1, t_2]$, the segment $\boldsymbol{x} = \boldsymbol{p}_0 + t\boldsymbol{d}$ is within the k-DOP.

4.3. Bounding k-DOP containment test

For collision detection, we would like to also know if a k-UDOP intersects with another k-DOP. For this, let the k-UDOP be defined by a $\boldsymbol{c}_1 \in \mathbb{R}^D$ center, $\boldsymbol{v}_i \in \mathbb{R}^D, \|\boldsymbol{v}_i\|_2 = 1$ directions and $h_i > 0$ distances, and the k-DOP be defined by the same $\boldsymbol{v}_i \in \mathbb{R}^D$ directions but with a $\boldsymbol{c}_2 \in \mathbb{R}^D$ center and $g_i >$ distances. Then, the k-DOP is inside the k-UDOP if the distance vector between the centers ($\boldsymbol{c}_2 - \boldsymbol{c}_1$) projected onto each $\boldsymbol{v}_i$ is less than the difference between the k-DOP distances ($h_i - g_i$). Algorithm 3 summarizes the above and allows efficient utilization of k-UDOP acceleration structures for collision detection tasks in any dimension.

Algorithm 3 Bounding k-DOP containment test

Input: $\boldsymbol{c}_1, \boldsymbol{c}_2$ centers, $\mathbf{v}_i$ common directions, h_i, g_i distances
Output: True only if first k-DOP contains the second
for all $\mathbf{v}_i$ **do**
  if $(\boldsymbol{c}_2 - \boldsymbol{c}_1)^{\mathrm{T}} \cdot \boldsymbol{v}_i \geq h_i - g_i$ **then**
    return ***false***
  end if
end for
return ***true***

5. Test results

We observed that the k-DOP does not always utilize all sides, so the generated polygon often has less than k number of edges. To find a reasonable choice of k, we generated random points around $\boldsymbol{c}$ from different distributions and measured various metrics of the resulting k-DOP. We tested 50 different scenes with k increasing from 3 to 300. The means of the various metrics are shown in Table 1.

We also noticed that as we increase k, the k-UDOP shape stabilizes, as if a fixed point solution was found. To verify this, we have taken the symmetric difference of the polygons of consecutive pairs of k-DOPs and measured the average difference of its area. Generally, the difference converged to zero; however, the area fluctuated even for large k values in a few cases. This seems to have been caused by empty

Table 1. Average values of different metrics measured from 50 different set of points for every k value between 3 and 300.

k	**Perimeter**	**Area**	**# of sides**	$\frac{\#\text{ of sides}}{k}$
3	3.217	0.911	3.00	100%
4	3.669	1.070	4.00	100%
5	3.896	1.194	4.42	88.4%
6	3.820	1.165	4.84	80.7%
7	4.018	1.242	4.96	70.9%
8	3.952	1.232	5.06	63.3%
9	3.906	1.264	5.14	57.1%
10	3.757	1.149	5.34	53.4%
11	3.875	1.232	5.70	51.8%
12	3.925	1.233	5.72	47.7%
⋮	⋮	⋮	⋮	⋮
298	4.033	1.279	15.74	5.28%
299	4.042	1.280	15.92	5.32%
300	4.024	1.266	15.86	5.29%

areas in-between the clusters of generated points, which placed at least one of the k-DOP sides very far from $\boldsymbol{c}$ for certain k values but was cut off in other cases. The results of the different tests are visualized in Fig. 4.

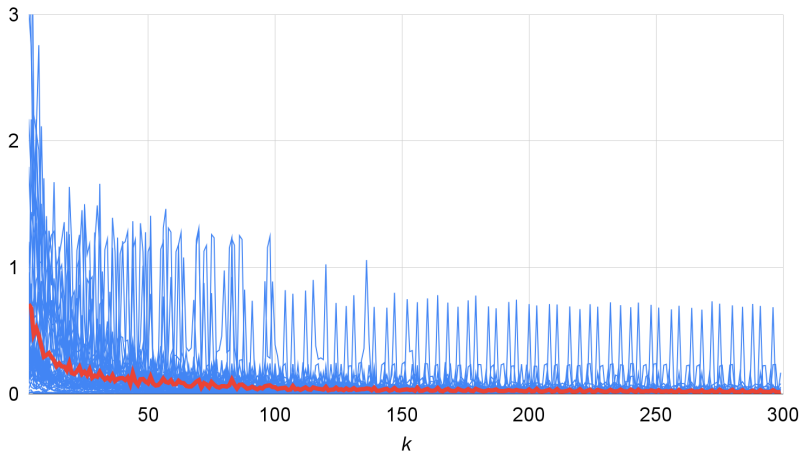


Figure 4. Measured convergence of consecutive polygons for 50 random cases (blue lines) and their average (red). This difference is measured in the area of the symmetric difference of the polygons generated from k and $k+1$-DOPs.

The convergence was expected, since if $\boldsymbol{w}_j$ are directions towards each object's closest point to $\boldsymbol{c}$, then drawing perpendicular lines through the footpoints to each $\boldsymbol{w}_j$, we obtain the convex polygon that the algorithm seems to approach. This is

because at least one of the directions $\boldsymbol{v}_i$ will generally get closer to each $\boldsymbol{w}_j$ direction, so the algorithm will choose a corresponding line that is almost perpendicular to $\boldsymbol{w}_j$.

6. Conclusions

We presented a simple and efficient algorithm to compute D-dimensional k-UDOPs for a prescribed position and a set of k fixed directions.

In the plane, we showed that the resulting convex polygon converges to a fixed shape empirically, whose number of effective sides stayed within 16, even for $k = 300$. As such, large k figures function more to orient the resulting k-UDOP.

Additionally, we presented conversion algorithms to polytope meshes in arbitrary dimensions for both bounding and k-UDOPs and direct ray intersection and bounding k-DOP containment tests.

References

[1] C. Bálint, G. Valasek, L. Gergó: *Operations on Signed Distance Functions*, Acta Cybernetica 24.1 (May 2019), pp. 17–28, doi: https://doi.org/10.14232/actacyb.24.1.2019.3, url: https://cyber.bibl.u-szeged.hu/index.php/actcybern/article/view/4004.

[2] S. Dinas, J. M. Bañón: *A literature review of bounding volumes hierarchy focused on collision detection*, Ingeniería y competitividad 17.1 (2015), pp. 49–62.

[3] C. Ericson: *Real-time collision detection*, CRC Press, 2004.

[4] G. Farin: *Curves and Surfaces for Computer Aided Geometric Design (3rd Ed.): A Practical Guide*, San Diego, CA, USA: Academic Press Professional, Inc., 1993, isbn: 0-12-249052-5.

[5] J. Hart: *Sphere Tracing: A Geometric Method for the Antialiased Ray Tracing of Implicit Surfaces*, The Visual Computer 12 (June 1995), doi: https://doi.org/10.1007/s003710050084.

[6] T. L. Kay, J. T. Kajiya: *Ray Tracing Complex Scenes*, in: Proceedings of the 13th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH '86, New York, NY, USA: Association for Computing Machinery, 1986, pp. 269–278, isbn: 0897911962, doi: https://doi.org/10.1145/15922.15916.

[7] J. Klosowski, M. Held, J. Mitchell, H. Sowizral, K. Zikan: *Efficient collision detection using bounding volume hierarchies of k-DOPs*, IEEE Transactions on Visualization and Computer Graphics 4.1 (1998), pp. 21–36, doi: https://doi.org/10.1109/2945.675649.

[8] G. Valasek, C. Bálint, A. Leitereg: *Footvector Representation of Curves and Surfaces*, Acta Cybernetica (Aug. 2021), doi: https://doi.org/10.14232/actacyb.290145, url: https://cyber.bibl.u-szeged.hu/index.php/actcybern/article/view/4205.

[9] G. Zachmann: *Rapid collision detection by dynamically aligned DOP-trees*, in: Proceedings. IEEE 1998 Virtual Reality Annual International Symposium (Cat. No.98CB36180), 1998, pp. 90–97, doi: https://doi.org/10.1109/VRAIS.1998.658428.

Annales Mathematicae et Informaticae
56 (2022) pp. 58–70
DOI: https://doi.org/10.33039/ami.2022.12.007
URL: https://ami.uni-eszterhazy.hu

Tree generating context-free grammars and regular tree grammars are equivalent

Dávid Kószó

Department of Foundations of Computer Science
University of Szeged
Szeged, Hungary
koszod@inf.u-szeged.hu

Abstract. We show that it is decidable whether the language generated by a given context-free grammar over a tree alphabet is a tree language. Furthermore, if the answer to this question is "yes", then we can even effectively construct a regular tree grammar which generates that tree language.

Keywords: context-free grammar, regular tree grammar, tree language, parenthesis grammar, tree generating context-free grammar, decidability

AMS Subject Classification: 68Q45

1. Introduction

Context-free grammars (for short: cfg) were introduced in [3] in order to describe the structure of sentences and words in natural languages. Since then, a beautiful theory of cfg has been evolved, *cf. e.g.* [6, 7]. In computer science cfg are used to describe the structure of programming languages and play a crucial role in the Document Type Definition (DTD) of the Extensible Markup Language (XML) as well. The language generated by a Γ-cfg G, *i.e.*, a cfg over some alphabet Γ, is denoted by $\mathrm{L}(G)$ and called a context-free language.

In order to define well-formed terms, we use a special alphabet called a ranked alphabet and three further special symbols. A ranked alphabet Σ is an alphabet in which we associate with each symbol a unique rank. The three special symbols are the opening angle bracket "$\langle$", the closing angle bracket "$\rangle$", and the symbol "#". The set of these special symbols is denoted by Ξ and the alphabet Σ^{Ξ} containing

Submitted: October 25, 2022
Accepted: December 3, 2022
Published online: December 28, 2022

the symbols of Σ and Ξ is called a tree alphabet. Using the three special symbols as separators, the Σ-terms are defined in the standard way, *i.e.*, each Σ-term is a string $\sigma\langle\xi_1\#\ldots\#\xi_k\rangle$ over the tree alphabet Σ^{Ξ}, where σ has rank k for some natural number k, and $\xi_1,\ldots,\xi_k$ are Σ-terms.

Since each Σ-term can be depicted as a tree-like directed labelled graph, we often refer to Σ-terms as Σ-trees. Moreover, a set of Σ-trees is called a (formal) Σ-tree language. We denote the set of all Σ-trees by T_Σ and we call a Σ^{Ξ}-cfg G tree generating if $\mathrm{L}(G) \subseteq \mathrm{T}_\Sigma$.

To generate Σ-tree languages, among others regular tree grammars (for short: Σ-rtg) were defined [2, 4, 5]. The Σ-tree language generated by a Σ-rtg $\mathcal{G}$, denoted by $\mathrm{L}(\mathcal{G})$, is called a regular Σ-tree language. The connection between context-free languages and regular tree languages has been thoroughly investigated. Among others, it was shown that, for each language L, the following statements are equivalent [2, 10]:

(1) L is a context-free language,
(2) L is the yield of a regular tree language.

Then several authors have exploited this strong connection, *cf.* *e.g.*, [4, 11, 12]. Furthermore, each Σ-rtg is evidently a tree generating Σ^{Ξ}-cfg. However, to the best of our knowledge, it has not been cleared yet whether there exists a Σ-tree language, which can be generated by a Σ^{Ξ}-cfg but it is not regular. Hence, here we deal with the following questions and answer them positively:

(Q1) Given a Σ^{Ξ}-cfg G, is it decidable whether G is tree generating?
(Q2) Given a Σ^{Ξ}-cfg G such that G is tree generating, is $\mathrm{L}(G)$ regular, and if yes, can we effectively construct a Σ-rtg $\mathcal{G}$ such that $\mathrm{L}(\mathcal{G}) = \mathrm{L}(G)$?

To answer the questions, we will consider the class of parenthesis grammars. A Γ-parenthesis grammar [9] is a Γ-cfg in which each rule has the form $A \to \langle\alpha\rangle$, where A is a nonterminal and α is a string over $\Gamma \setminus \{\langle, \rangle\}$. A language generated by a Γ-parenthesis grammar is called a Γ-parenthesis language. Interestingly, we can give a transduction φ such that, for each Σ-tree language L, the language $\varphi(L)$ is a Σ^{Ξ}-parenthesis language. (We note that there exists a Σ^{Ξ}-parenthesis language, which is not an image of any Σ-tree language under φ.) We prove our results by exploiting this connection between Σ-rtg and Σ^{Ξ}-parenthesis grammars and by applying Knuth's results [8]:

(R1) it is decidable, for a given Γ-cfg G, whether $\mathrm{L}(G)$ is a Γ-parenthesis language and
(R2) for a given Γ-cfg G such that $\mathrm{L}(G)$ is a Γ-parenthesis language, we can effectively construct a Γ-parenthesis grammar G' such that $\mathrm{L}(G') = \mathrm{L}(G)$.

We mention that, for unranked trees, question (Q1) was answered positively in [1].

Our paper is organized as follows. In Section 2 we recall the necessary notions and notations. In Section 3 we recall the concept of cfg and of rtg, and results on parenthesis grammars. In Section 4 we recall the concept of sequential transducer, which will be useful to prove our results. Finally, in Section 5 we give our results.

2. Preliminaries

2.1. Basic concepts

We denote the set $\{0, 1, 2, \ldots\}$ of nonnegative integers by $\mathbb{N}$ and we let $\mathbb{N}_+ = \mathbb{N}\backslash\{0\}$. For every $k \in \mathbb{N}$, we let $[k] = \{i \in \mathbb{N}_+ \mid i \leq k\}$. In particular, $[0] = \emptyset$. Furthermore, we denote the set of integers by $\mathbb{Z}$.

Let A be a set and $R, S \subseteq A \times A$ binary relations. The *composition of R and S*, denoted by $R \circ S$, is the set

$$R \circ S = \{(a, c) \in A \times A \mid (\exists b \in A) : (a, b) \in R \wedge (b, c) \in S\} \ .$$

We define, for each $n \in \mathbb{N}$, the *n-fold composition of R*, denoted by R^n, by $R^0 = \{(a, a) \mid a \in A\}$ and by $R^n = R^{n-1} \circ R$ for each $n \in \mathbb{N}_+$.

2.2. Strings and trees

We assume that the reader is familiar with the fundamental concepts and results of the theory formal languages [6, 7], and also of tree languages [4, 5].

An *alphabet* is a finite set. Let Γ be an alphabet. A *string (over Γ)* is a finite sequence $a_1 \cdots a_k$ with $k \in \mathbb{N}$ and $a_i \in \Gamma$ for each $i \in [k]$. The *length of* $a_1 \cdots a_k$, denoted by $\mathrm{len}(a_1 \cdots a_k)$, is defined in the standard way. We denote by Γ^* the *set of all strings over* Γ and by ε the *empty string*. Each subset $L \subseteq \Gamma^*$ is called a *language over* Γ. Moreover, for all $v, w \in \Gamma^*$, we denote by vw the *concatenation of v and w*, and the *set of prefixes of v*, denoted by $\mathrm{prefix}(v)$, is defined by $\mathrm{prefix}(v) = \{u \in \Gamma^* \mid (\exists v' \in \Gamma^*) : v = uv'\}$.

A *ranked alphabet* is a tuple (Σ, rk), where Σ is an alphabet and $\mathrm{rk} : \Sigma \to \mathbb{N}$ is a mapping, called *rank mapping*, such that $\mathrm{rk}^{-1}(0) \neq \emptyset$. For all $k \in \mathbb{N}$, we let

$$\Sigma^{(k)} = \{\sigma \in \Sigma \mid \mathrm{rk}(\sigma) = k\} \ .$$

We always abbreviate (Σ, rk) by Σ.

Next we define Σ-trees. In the literature, Σ-trees are defined by using the opening and the closing parenthesis "(" and ")", respectively, and the comma "," as separators [4, 5]. In this paper, we will focus on these separators in trees frequently. Since it is easy to confuse these separators with the two parentheses in other formulas, we intentionally deviate and use the opening and the closing angle brackets "$\langle$" and "$\rangle$", respectively, and the symbol "#" to define Σ-trees.

Let Ξ be the set which consists of "$\langle$" and "$\rangle$" and "#". A *tree alphabet* Σ^{Ξ} is an alphabet consisting of symbols of Σ and Ξ, *i.e.*, $\Sigma^{\Xi} = \Sigma \cup \Xi$.

Let H be a set such that $H \cap \Sigma^{\Xi} = \emptyset$. The *set of Σ-trees over H*, denoted by $\mathrm{T}_\Sigma(H)$, is the smallest set $T \subseteq (\Sigma^{\Xi} \cup H)^*$ such that

(i) $H \subseteq T$ and

(ii) if $k \in \mathbb{N}$, $\sigma \in \Sigma^{(k)}$, and $\xi_1, \ldots, \xi_k \in T$, then $\sigma\langle \xi_1 \# \ldots \# \xi_k \rangle \in T$.

We abbreviate $\mathrm{T}_\Sigma(\emptyset)$ by T_Σ. A *Σ-tree language* (or just: *tree language*) is a subset of T_Σ.

From now on, we let Σ be an arbitrary ranked alphabet if not specified otherwise.

3. Grammar models

3.1. Context-free grammars

Let Γ be an alphabet. A *context-free grammar over* Γ (for short: Γ-*cfg*) [6, 7] is a triple $G = (N, S, R)$ where N is a finite set (*nonterminals*) with $N \cap \Gamma = \emptyset$, $S \in N$ (*start symbol*), and R is a finite set (*rules*); each rule has the form $A \to \alpha$ where $A \in N$ and α is a string over $N \cup \Gamma$, *i.e.*, $\alpha \in (N \cup \Gamma)^*$. Furthermore, we call each element $a \in \Gamma$ a *terminal*.

Let $G = (N, S, R)$ be a Γ-cfg and let $r = (A \to \alpha)$ be a rule. We call A and α the *left-hand side* and the *right-hand side* of r, respectively. Moreover, we call r a *chain rule* (an ε-*rule*) if $\alpha \in N$ (if $\alpha = \varepsilon$, respectively). We say that G is *chain-free* (ε-*free*) if G does not have chain rules (ε-rules, respectively).

The *(leftmost) derivation relation* $\Rightarrow_G$ is defined such that, for every $u \in \Gamma^*$, $\gamma \in (N \cup \Gamma)^*$, and rule $A \to \alpha$ in R, we have $uA\gamma \Rightarrow_G u\alpha\gamma$. If G is clear from the context, then we abbreviate $\Rightarrow_G$ by $\Rightarrow$. For all $\gamma, \omega \in (N \cup \Gamma)^*$, if $\gamma \Rightarrow^n \omega$ for some $n \in \mathbb{N}$, then we say that this derivation has length n. As usual, we denote the reflexive and transitive closure of $\Rightarrow$ by $\Rightarrow^*$, *i.e.*, $\Rightarrow^* = \bigcup_{n \in \mathbb{N}} \Rightarrow^n$.

The *language generated by* G is the set

$$\mathrm{L}(G) = \{w \in \Gamma^* \mid S \Rightarrow^* w\} \ .$$

For each $L \subseteq \Gamma^*$, we call L a *context-free language* if there exists a Γ-cfg G such that $\mathrm{L}(G) = L$.

We call a nonterminal $A \in N$ *useful (in* G*)* if there exist $u, w \in \Gamma^*$ and $\gamma \in (N \cup \Gamma)^*$ such that $S \Rightarrow^* uA\gamma \Rightarrow^* w$. Moreover, if every $A \in N$ is useful, then we call G *reduced* [6, p. 78].

Lemma 3.1. [6, Thm. 3.2.3] If G is a Γ-cfg, then we can effectively construct a reduced Γ-cfg $\widehat{G}$ such that $\mathrm{L}(\widehat{G}) = \mathrm{L}(G)$.

Next we define parenthesis grammars and languages. They are normally defined by using the opening and the closing parenthesis "(" and ")". Later, in Section 5, we will relate Σ-tree languages and parenthesis languages. Therefore, we will consistently deviate from the convention and use the angle brackets "$\langle$" and "$\rangle$" instead of the usual "(" and ")", respectively; however we keep the notions parenthesis grammar and parenthesis language.

In the rest of this section, we let Γ be an alphabet which contains the angle brackets "$\langle$" and "$\rangle$".

A Γ-*parenthesis grammar* [8] (or just: parenthesis grammar) is a Γ-cfg $G = (N, S, R)$ such that each rule in R has the form $A \to \langle\theta\rangle$ with $\theta \in (N \cup \Gamma \setminus \{\langle, \rangle\})^*$.

Table 1. Illustration of the content and the deficiency mappings, and the notion balanced.

w	$c(w)$	$d(w)$	balanced
$a\langle b\langle\rangle\rangle$	0	0	yes
$\langle\rangle\rangle$	-1	1	no
$\langle a\langle$	2	0	no
$\langle a\rangle b\rangle\rangle\langle b\rangle$	-2	2	no
$\rangle\rangle\rangle\langle\langle\langle$	0	3	no

Observation 3.2. If G is a Γ-parenthesis grammar, then G is chain-free and ε-free.

We call a language $L \subseteq \Gamma^*$ a Γ-*parenthesis language* (or just: parenthesis language) if there exists a Γ-parenthesis grammar G such that $\mathrm{L}(G) = L$.

Here we draw attention to the following phenomenon. Let G be a Γ-cfg such that $\mathrm{L}(G)$ is a parenthesis language. Then it does not follow that G is a parenthesis grammar. Rather, it follows that there exists a Γ-parenthesis grammar G' such that $\mathrm{L}(G') = \mathrm{L}(G)$. We will use this fact later.

The *content mapping* $c : \Gamma^* \to \mathbb{Z}$ and the *deficiency mapping* $d : \Gamma^* \to \mathbb{N}$ [8] are defined, for each $w \in \Gamma^*$, as follows:

(i) if $w = \varepsilon$, then we let $c(\varepsilon) = d(\varepsilon) = 0$,

(ii) if $w = a$ for some $a \in \Gamma$, then we let

$$c(a) = \begin{cases} 1 & \text{if } a = \langle \\ -1 & \text{if } a = \rangle \\ 0 & \text{otherwise} \end{cases} \quad \text{and} \quad d(a) = \begin{cases} 1 & \text{if } a = \rangle \\ 0 & \text{otherwise} \end{cases}, \quad \text{and}$$

(iii) if $w = va$ with $v \in \Gamma^*$ and $a \in \Gamma$, then we let $c(va) = c(v) + c(a)$ and $d(va) = \max\{d(v), d(a) - c(v)\}$.

Intuitively, for each string $w \in \Gamma^*$, the values $c(w)$ and $d(w)$ show us the excess of left parentheses over right parentheses in w and the greatest deficiency of left parentheses from right parentheses in any prefix of w, respectively. A string $w \in \Gamma^*$ is *balanced* if $c(w) = d(w) = 0$, and furthermore, a language $L \subseteq \Gamma^*$ is *balanced* if every $w \in L$ is balanced.

Observe that, for all balanced $u, v \in \Gamma^*$, also uv is balanced. Furthermore, each $u \in (\Gamma \setminus \{\langle, \rangle\})^*$ is balanced as well.

Example 3.3. Let $\Gamma = \{a, b, \langle, \rangle\}$. Table 1 shows, for some $w \in \Gamma^*$, the values of the content and the deficiency mappings, *i.e.*, $c(w)$ and $d(w)$, respectively, and whether w is balanced or not.

The next lemma shows an important property of parenthesis grammars and it will be useful to prove the results in Section 5.

Lemma 3.4. Let $G = (N, S, R)$ be a Γ-parenthesis grammar. Furthermore, let $A \in N$ and $w \in \Gamma^*$. If $A \Rightarrow^* w$, then $w = \langle u \rangle$ for some $u \in \Gamma^*$ such that u is balanced.

Proof. We prove our statement by induction on the length of the derivation $A \Rightarrow^* w$. Assume that $A \Rightarrow w$. Then, since G is a parenthesis grammar, we have $w = \langle u \rangle$ for some $u \in (\Gamma \setminus \{\langle, \rangle\})^*$. Hence u is balanced.

Now assume that $A \Rightarrow^{n+1} w$ for some $n \in \mathbb{N}_+$. Then, since G is a parenthesis grammar, there exist $k \in \mathbb{N}_+$, $v_0, v_1, v_2, \ldots, v_k$ in $(\Gamma \setminus \{\langle, \rangle\})^*$, $A_1, A_2, \ldots, A_k \in N$, $n_1, n_2, \ldots, n_k \in [n]$, and $w_1, w_2, \ldots, w_k \in \Gamma^*$ such that

- $w = \langle v_0 w_1 v_1 w_2 v_2 \cdots w_k v_k \rangle$,
- $A \to \langle v_0 A_1 v_1 A_2 v_2 \cdots A_k v_k \rangle$ is in R,
- for each $i \in [k]$ we have $A_i \Rightarrow^{n_i} w_i$,
- $n_1 + n_2 + \ldots + n_k = n$, and
- we have

$$A \Rightarrow^1 \langle v_0 A_1 v_1 A_2 v_2 \cdots A_k v_k \rangle \Rightarrow^{n_1} \langle v_0 w_1 v_1 A_2 v_2 \cdots A_k v_k \rangle \Rightarrow^{n_2} \cdots \Rightarrow^{n_k} w \ .$$

By I.H., for each $i \in [k]$, we may assume that there exists $u_i \in \Gamma^*$ such that $w_i = \langle u_i \rangle$ and u_i is balanced. Thus, for $u = v_0 w_1 v_1 w_2 v_2 \cdots w_k v_k$ it holds that $w = \langle u \rangle$ and u is balanced. This completes our proof. □

Let $w \in \Gamma^*$. For every $a, b \in \Gamma$, the terminals a, b are called *associates (in w)* [8] if $w = uavbv'$ for some $u, v, v' \in \Gamma^*$ and vb is balanced. A language $L \subseteq \Gamma^*$ is said to have *bound associates* if there exists a constant $K \in \mathbb{N}_+$ such that for all $w = uav$ in L with $u, v \in \Gamma^*$ and $a \in \Gamma$, the terminal a has at most K associates in w.

Example 3.5. Let $\Gamma = \{a, b, \langle, \rangle\}$. We consider the Γ-cfg

$$G = (\{S\}, S, \{\ S \to \varepsilon\ ,\ S \to aSb\ \})\ .$$

Then we have $\mathrm{L}(G) = \{a^n b^n \mid n \in \mathbb{N}\}$. Moreover, G is obviously not a parenthesis grammar. Now we consider the Γ-cfg

$$G' = (\{S'\}, S', \{\ S' \to \langle\rangle\ ,\ S' \to \langle aS'b \rangle\ \})\ .$$

Then, for each $n \in \mathbb{N}$, we have

$$S' \Rightarrow_{G'} \langle aS'b \rangle \Rightarrow^*_{G'} \langle a\langle \cdots \langle aS'b \rangle \cdots \rangle b \rangle \Rightarrow_{G'} \langle a\langle \cdots \langle a\langle\rangle b \rangle \cdots \rangle b \rangle\ ,$$

where both a and b occur n times. In particular, $\mathrm{L}(G')$ contains the string "$\langle\rangle$". Clearly, G' is a parenthesis grammar, and $\mathrm{L}(G')$ is balanced and has bounded associates.

Lemma 3.6. [8, Cor. 4] It is decidable, for arbitrary Γ-cfg G_1 and parenthesis grammar G_2, whether $\mathrm{L}(G_1) \subseteq \mathrm{L}(G_2)$.

Theorem 3.7. *cf.* [8, Thm. 4] The following statements hold true.

1. A context-free language is balanced and has bounded associates iff it is a parenthesis language.
2. For each Γ-cfg G, if $\mathrm{L}(G)$ is a parenthesis language, then we can effectively construct a Γ-parenthesis grammar G' such that $\mathrm{L}(G') = \mathrm{L}(G)$.

The next result is an easy consequence of Theorem 3.7(1).

Corollary 3.8. Let G be a Γ-parenthesis grammar and $L \subseteq \mathrm{L}(G)$ a context-free language. Then L is a parenthesis language.

Proof. Since $\mathrm{L}(G)$ is a parenthesis language, by Theorem 3.7(1), $\mathrm{L}(G)$ is balanced and has bounded associates. Clearly, also L is balanced. Moreover, since $\mathrm{L}(G)$ has bounded associates, there exists a constant $K \in \mathbb{N}_+$ such that for all $w = uav$ in $\mathrm{L}(G)$ with $u, v \in \Gamma^*$ and $a \in \Gamma$, the terminal a has at most K associates. Since $L \subseteq \mathrm{L}(G)$, for all $w = uav$ in L with $u, v \in \Gamma^*$ and $a \in \Gamma$, the terminal a has at most K associates, *i.e.*, also L has bounded associates. Hence, by Theorem 3.7(1), L is a parenthesis language as well. □

Now we define a new subclass of context-free grammars, which we call tree generating context-free grammars. Formally, for each Σ^{Ξ}-cfg G, we say that G is *tree generating* if $\mathrm{L}(G) \subseteq \mathrm{T}_\Sigma$.

In the next example we give a tree generating Σ^{Ξ}-cfg.

Example 3.9. Let $\Sigma = \{\omega^{(3)}, \beta^{(0)}\}$. We consider the Σ^{Ξ}-cfg

$$G = (\{S, A, B, C\}, S, R) \ ,$$

where

$$R = \{\ S \to ASB\rangle \ ,\ S \to ACB\rangle \ ,\ A \to \omega\langle C\# \ ,\ B \to \#C \ ,\ C \to \beta\langle\rangle \ \} \ .$$

Then we have, *e.g.*,

$$\begin{aligned} S &\Rightarrow ASB\rangle \Rightarrow \omega\langle C\#SB\rangle \Rightarrow \omega\langle\beta\langle\rangle\#SB\rangle \\ &\Rightarrow \omega\langle\beta\langle\rangle\#ACB\rangle B\rangle \Rightarrow \omega\langle\beta\langle\rangle\#\omega\langle C\#CB\rangle B\rangle \\ &\Rightarrow \omega\langle\beta\langle\rangle\#\omega\langle\beta\langle\rangle\#CB\rangle B\rangle \Rightarrow \omega\langle\beta\langle\rangle\#\omega\langle\beta\langle\rangle\#\beta\langle\rangle B\rangle B\rangle \\ &\Rightarrow \omega\langle\beta\langle\rangle\#\omega\langle\beta\langle\rangle\#\beta\langle\rangle\#C\rangle B\rangle \Rightarrow \omega\langle\beta\langle\rangle\#\omega\langle\beta\langle\rangle\#\beta\langle\rangle\#\beta\langle\rangle\rangle B\rangle \\ &\Rightarrow \omega\langle\beta\langle\rangle\#\omega\langle\beta\langle\rangle\#\beta\langle\rangle\#\beta\langle\rangle\rangle\#C\rangle \Rightarrow \omega\langle\beta\langle\rangle\#\omega\langle\beta\langle\rangle\#\beta\langle\rangle\#\beta\langle\rangle\rangle\#\beta\langle\rangle\rangle \ . \end{aligned}$$

Evidently, $\mathrm{L}(G) \subseteq \mathrm{T}_\Sigma$, hence G is tree generating.

3.2. Regular tree grammars

A *regular tree grammar over* Σ (for short: Σ-*rtg*) [2, 4, 5] is a Σ^{Ξ}-cfg $\mathcal{G} = (N, S, R)$ such that each rule in R has the form $A \to \eta$ with $\eta \in \mathrm{T}_\Sigma(N)$. Obviously, if $A \Rightarrow^* \xi$ for some $\xi \in (\Sigma^{\Xi})^*$, then $\xi \in \mathrm{T}_\Sigma$.

The Σ-*tree language generated by* $\mathcal{G}$ is the set

$$\mathrm{L}(\mathcal{G}) = \{\xi \in \mathrm{T}_\Sigma \mid S \Rightarrow^* \xi\} \ .$$

We call each $L \subseteq \mathrm{T}_\Sigma$ *regular* if there exists a Σ-rtg $\mathcal{G}$ such that $\mathrm{L}(\mathcal{G}) = L$. Observe that each Σ-rtg is a tree generating context-free grammar.

Example 3.10. Let $\Sigma = \{\omega^{(3)}, \beta^{(0)}\}$. We consider the Σ-rtg $\mathcal{G} = (\{S\}, S, R)$, where $R = \{\ S \to \omega\langle\beta\langle\rangle\#\beta\langle\rangle\#\beta\langle\rangle\rangle\ ,\ S \to \omega\langle\beta\langle\rangle\#S\#\beta\langle\rangle\rangle\ \}$. Fig. 1 shows, for each $n \in \mathbb{N}_+$, the tree ξ_n and the derivation of $\mathcal{G}$ for ξ_n. In fact, $\mathrm{L}(\mathcal{G}) = \{\xi_n \mid n \in \mathbb{N}_+\}$. One can show that, for the tree generating Σ^Ξ-cfg G defined in Example 3.9, we have $\mathrm{L}(\mathcal{G}) = \mathrm{L}(G)$.

4. Sequential transducers

To prove our results in the next section, it is necessary to recall the concept of sequential transducer and the Sequential Transducer Theorem.

Let Γ and Δ be two alphabets. A (Γ, Δ)-*sequential transducer* (or just sequential transducer) [6] is a tuple $\mathcal{S} = (Q, q_0, \delta)$ where Q is a finite nonempty set (*states*), $q_0 \in Q$ (*start state*), and δ is a finite subset of $Q \times \Gamma^* \times \Delta^* \times Q$ (*transitions*).

Let $\mathcal{S} = (Q, q_0, \delta)$ be a (Γ, Δ)-sequential transducer. For all $w \in \Gamma^*$ and $u \in \Delta^*$, we have $u \in \mathcal{S}(w)$ iff there exist $k \in \mathbb{N}$, $w_1, \ldots, w_k \in \Gamma^*$, $u_1, \ldots, u_k \in \Delta^*$, and $q_1, \ldots, q_k \in Q$ such that $w = w_1 \cdots w_k$, $u = u_1 \cdots u_k$, and $(q_{i-1}, w_i, u_i, q_i) \in \delta$ for each $i \in [k]$. Moreover, for every $L \subseteq \Gamma^*$, we have

$$\mathcal{S}(L) = \bigcup_{w \in L} \mathcal{S}(w) \ .$$

We call a binary relation $\varphi \subseteq \Gamma^* \times \Delta^*$ a (Γ, Δ)-*transduction* (or just: transduction) if there exists a (Γ, Δ)-sequential transducer $\mathcal{S}$ such that $\mathcal{S}(w) = \varphi(w)$ for every $w \in \Gamma^*$.

Lemma 4.1. [6, Thm. 6.4.3] (The Sequential Transducer Theorem) Let $L \subseteq \Gamma^*$ be a context-free language and $\mathcal{S}$ be a (Γ, Δ)-sequential transducer. Then $\mathcal{S}(L) \subseteq \Delta^*$ is a context-free language as well.

5. Results

In this section we answer questions (Q1) and (Q2), which we proposed in the Introduction. To answer these questions the following steps are necessary.

Let $\varphi : (\Sigma^\Xi)^* \to (\Sigma^\Xi)^*$ be the mapping such that, for each string $w \in (\Sigma^\Xi)^*$, the mapping φ replaces every occurrence of $\sigma\langle$ in w into $\langle\sigma$ simultaneously for all $\sigma \in \Sigma$. Formally, for every string

$$w = v_0\sigma_1\langle v_1 \cdots \sigma_k\langle v_k \text{ over } \Sigma^\Xi$$

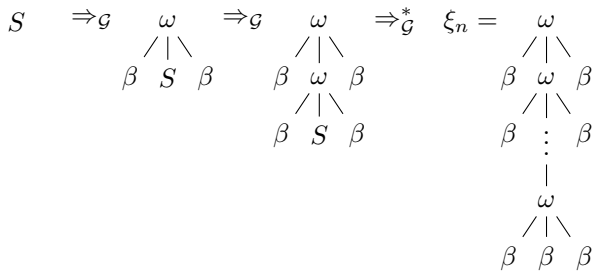


Figure 1. A derivation of the Σ-rtg $\mathcal{G}$ defined in Example 3.10 for $n \in \mathbb{N}_+$ and ξ_n, where ξ_n is the tree in which the symbol ω occurs n times.

Table 2. The illustration of the mapping φ.

w	$\varphi(w)$
$\omega\langle\beta\langle\rangle\#\beta\langle\rangle\#\beta\langle\rangle\rangle$	$\langle\omega\langle\beta\rangle\#\langle\beta\rangle\#\langle\beta\rangle\rangle$
$\omega\beta\langle\rangle\omega\langle\#$	$\omega\langle\beta\rangle\langle\omega\#$
$\langle\langle\rangle\rangle\langle\rangle$	$\langle\langle\rangle\rangle\langle\rangle$
$\langle\langle\#\rangle\langle\langle\langle\#\rangle$	$\langle\langle\#\rangle\langle\langle\langle\#\rangle$

with $k \in \mathbb{N}$, $v_0, v_1, \ldots, v_k \in (\Sigma^{\Xi})^*$, $\sigma_1, \ldots, \sigma_k \in \Sigma$ such that, for each $i \in \{0, \ldots, k\}$, there do not exist $u, v \in (\Sigma^{\Xi})^*$ and $\sigma \in \Sigma$ such that $v_i = u\sigma\langle v$, we have

$$\varphi(w) = v_0\langle\sigma_1 v_1 \cdots \langle\sigma_k v_k \ .$$

Example 5.1. Let $\Sigma = \{\omega^{(3)}, \beta^{(0)}\}$. Table 2 shows $\varphi(w)$ for some particular w over Σ^{Ξ}.

Now we give a $(\Sigma^{\Xi}, \Sigma^{\Xi})$-sequential transducer $\mathcal{S}$ such that, for all strings w over Σ^{Ξ}, we have $\mathcal{S}(w) = \varphi(w)$. Fig. 2 depicts that sequential transducer $\mathcal{S} = (\{p, q\}, p, \delta)$ as follows. We represent every state $q' \in \{p, q\}$ as a circle with q' in its center, the start state p by an ingoing directed edge with the label "start", and each transition $(p', u, v, q') \in \delta$ by a directed edge from p' to q' with the label u/v. In order to make our figure compact, we add the quantifications "$(\forall\sigma \in \Sigma)$:", "$(\forall a \in \Xi)$:", or "$(\forall a \in \Xi \setminus \{\langle\})$:" to omit a few edges. Furthermore, the label of the edge from q to p consists of two lines representing concisely that $(q, \sigma\langle, \langle\sigma, p) \in \delta$ for every $\sigma \in \Sigma$ and $(q, a, a, p) \in \delta$ for each a in $\Xi \setminus \{\langle\}$, respectively. Observe that, for each w over Σ^{Ξ}, the set $\mathcal{S}(w)$ is a singleton set, and thus, we sometimes identify $\mathcal{S}(w)$ with its one and only element.

The following result shows that φ is a $(\Sigma^{\Xi}, \Sigma^{\Xi})$-transduction.

Lemma 5.2. For each w over Σ^{Ξ}, we have $\mathcal{S}(w) = \varphi(w)$.

Proof. We prove our statement by induction on the length of w. Clearly, for each w in $\Sigma^{\Xi} \cup \{\varepsilon\} \cup \{\sigma\langle \mid \sigma \in \Sigma\}$, we have $\mathcal{S}(w) = \varphi(w)$.

Now let $w = w'b$ for some $w' \in (\Sigma^{\Xi})^*$ and $b \in \Sigma^{\Xi}$. By I.H., we may assume that $\mathcal{S}(w') = \varphi(w')$. By the construction of $\mathcal{S}$, there exist $k \in \mathbb{N}_+$ and $w_1, \ldots, w_k \in (\Sigma^{\Xi})^*$ such that $w' = w_1 \cdots w_k$ and $1 \leq \mathrm{len}(w_i) \leq 2$ for all $i \in [k]$. Furthermore, there exist $u_1, \ldots, u_k \in (\Sigma^{\Xi})^*$ and $q_0, q_1, \ldots, q_k \in \{p, q\}$ such that $\mathcal{S}(w') = u_1 \cdots u_k$, $q_0 = p$, and $(q_{i-1}, w_i, u_i, q_i) \in \delta$ for each $i \in [k]$. We consider the next cases.

Assume that $w_k = a\sigma$ for some a in $\Sigma^{\Xi} \cup \{\varepsilon\}$ and $\sigma \in \Sigma$ and $b = \langle$. We have $(q_{k-1}, a, a, q) \in \delta$ if $a \in \Sigma$; and $(q_{k-1}, a, a, p) \in \delta$ if ($a \in \Xi$ and $q_{k-1} = p$) or ($a \in \Xi \setminus \{\langle\}$ and $q_{k-1} = q$). Since $\mathcal{S}(w') = \varphi(w')$, we may assume that $q_{k-1} \neq q$ or $a \neq \langle$. Moreover, both $(p, \sigma\langle, \langle\sigma, p) \in \delta$ and $(q, \sigma\langle, \langle\sigma, p) \in \delta$. Hence, $\mathcal{S}(w) = u_1 \cdots u_{k-1} a\langle\sigma$, and furthermore, $\mathcal{S}(w) = \varphi(w)$.

Otherwise, *i.e.*, $w_k \neq a\sigma$ or $b \neq \langle$, we have $(q_k, b, b, q') \in \delta$ for some $q' \in \{p, q\}$, and thus, $\mathcal{S}(w) = \varphi(w)$. □

The next result is an immediate consequence of Lemma 4.1 using the $(\Sigma^{\Xi}, \Sigma^{\Xi})$-sequential transducer $\mathcal{S}$ given at the beginning of this section.

Corollary 5.3. Let G be a Σ^{Ξ}-cfg. There exists a Σ^{Ξ}-cfg $G_{\mathcal{S}}$ such that $\mathrm{L}(G_{\mathcal{S}}) = \mathcal{S}(\mathrm{L}(G))$.

Next we show that $\mathcal{S}(\mathrm{T}_\Sigma)$ is a parenthesis language by constructing a Σ^{Ξ}-parenthesis grammar G_Σ such that $\mathrm{L}(G_\Sigma) = \mathcal{S}(\mathrm{T}_\Sigma)$. Let $G_\Sigma = (\{S\}, S, R)$ be the Σ^{Ξ}-cfg such that

$$R = \{S \to \langle \sigma \underbrace{S\#S\#\ldots\#S}_{k\text{-times}}\rangle \mid k \in \mathbb{N}, \sigma \in \Sigma^{(k)}\} \ .$$

Clearly, G_Σ is a parenthesis grammar.

Lemma 5.4. $\mathrm{L}(G_\Sigma) = \mathcal{S}(\mathrm{T}_\Sigma)$.

Proof. It is sufficient to prove that, for each $w \in (\Sigma^{\Xi})^*$, the following statements are equivalent.

1. $S \Rightarrow^*_{G_\Sigma} w$.
2. There exists $\xi \in \mathrm{T}_\Sigma$ such that $w = \mathcal{S}(\xi)$.

(1 ⇒ 2). We prove it by induction on the length of the derivation. If $S \Rightarrow_{G_\Sigma} w$, then $w = \langle\alpha\rangle$ for some $\alpha \in \Sigma^{(0)}$, and, clearly, for $\xi = \alpha\langle\rangle$, we have $\langle\alpha\rangle = \mathcal{S}(\alpha\langle\rangle)$.

Now assume that $S \Rightarrow^{n+1}_{G_\Sigma} w$ for some $n \in \mathbb{N}$. This derivation can be written in the form

$$S \Rightarrow_{G_\Sigma} \langle\sigma S\#S\#\ldots\#S\rangle \Rightarrow^*_{G_\Sigma} \langle\sigma w_1\#w_2\#\ldots\#w_k\rangle = w \ ,$$

where $S \to \langle\sigma S\#S\#\ldots\#S\rangle$ is in R, and by I.H., for each $i \in [k]$, there exists $\xi_i \in \mathrm{T}_\Sigma$ such that $w_i = \mathcal{S}(\xi_i)$. Then, for the tree $\xi = \sigma\langle\xi_1\#\xi_2\#\ldots\#\xi_k\rangle$, we have $w = \mathcal{S}(\xi)$.

(2 ⇒ 1). We prove it by structural induction on ξ. If $\xi = \alpha\langle\rangle$ for some $\alpha \in \Sigma^{(0)}$, then $w = \langle\alpha\rangle$. Since $S \to \langle\alpha\rangle$ is in R, we have $S \Rightarrow_{G_\Sigma} w$.

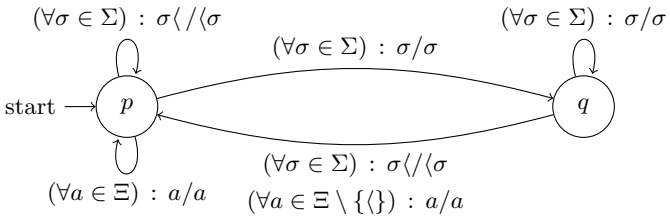


Figure 2. Illustration of the $(\Sigma^{\Xi}, \Sigma^{\Xi})$-sequential transducer $\mathcal{S}$ given at the beginning of Section 5.

Now let $\xi = \sigma\langle\xi_1\#\xi_2\#\ldots\#\xi_k\rangle$ for some $k \in \mathbb{N}_+$, $\sigma \in \Sigma^{(k)}$, and $\xi_1, \xi_2, \ldots, \xi_k \in \mathrm{T}_\Sigma$. Observe that we have $\mathcal{S}(\xi) = \langle\sigma\mathcal{S}(\xi_1)\#\mathcal{S}(\xi_2)\#\ldots\#\mathcal{S}(\xi_k)\rangle$. By I.H., for each $i \in [k]$, we have $S \Rightarrow^*_{G_\Sigma} \mathcal{S}(\xi_i)$. Since the rule $S \to \langle\sigma S\#S\#\ldots\#S\rangle$ is in R, we have

$$S \Rightarrow_{G_\Sigma} \langle\sigma S\#S\#\ldots\#S\rangle \Rightarrow^*_{G_\Sigma} \langle\sigma\mathcal{S}(\xi_1)\#\mathcal{S}(\xi_2)\#\ldots\#\mathcal{S}(\xi_k)\rangle = \mathcal{S}(\xi) = w \ .$$

□

Now we are ready to answer question (Q1) as follows.

Theorem 5.5. *It is decidable, for an arbitrary Σ^{Ξ}-cfg G, whether G is tree generating.*

Proof. By Corollary 5.3, there exists a Σ^{Ξ}-cfg $G_{\mathcal{S}}$ such that $\mathrm{L}(G_{\mathcal{S}}) = \mathcal{S}(\mathrm{L}(G))$. Then we have

$$\mathrm{L}(G) \subseteq \mathrm{T}_\Sigma \quad \text{iff} \quad \mathcal{S}(\mathrm{L}(G)) \subseteq \mathcal{S}(\mathrm{T}_\Sigma) \quad \text{iff} \quad \mathrm{L}(G_{\mathcal{S}}) \subseteq \mathrm{L}(G_\Sigma), \tag{5.1}$$

where the second equivalence follows from Lemma 5.4. By Lemma 3.6 (for $G_1 = G_{\mathcal{S}}$ and $G_2 = G_\Sigma$), it is decidable whether $\mathrm{L}(G_{\mathcal{S}}) \subseteq \mathrm{L}(G_\Sigma)$. Hence, by (5.1), it is decidable whether $\mathrm{L}(G) \subseteq \mathrm{T}_\Sigma$ as well. □

Built upon the preceding result, we give an answer to question (Q2).

Theorem 5.6. *Let G be a Σ^{Ξ}-cfg such that G is tree generating. We can effectively construct a Σ-rtg $\mathcal{G}$ such that $\mathrm{L}(\mathcal{G}) = \mathrm{L}(G)$.*

Proof. If G is a Σ-rtg, then we let $\mathcal{G} = G$ and we are done, otherwise we proceed as follows.

By Corollary 5.3, there exists a Σ^{Ξ}-cfg $G_{\mathcal{S}}$ such that $\mathrm{L}(G_{\mathcal{S}}) = \mathcal{S}(\mathrm{L}(G))$. Moreover, by (5.1), we have $\mathrm{L}(G_{\mathcal{S}}) \subseteq \mathrm{L}(G_\Sigma)$.

Since $\mathrm{L}(G_\Sigma)$ is a parenthesis language, by Corollary 3.8, also $\mathrm{L}(G_{\mathcal{S}})$ is a parenthesis language. By Theorem 3.7(2), we can effectively construct a Σ^{Ξ}-parenthesis grammar $G' = (N', S', R')$ such that $\mathrm{L}(G') = \mathrm{L}(G_{\mathcal{S}})$. Recall that, since G' is a parenthesis grammar, each rule in R' has the form $A \to \langle\theta\rangle$ such that $A \in N'$ and

θ is a string over $N' \cup \Sigma \cup \{\#\}$. We note that, by Observation 3.2, G' is chain-free and ε-free. Furthermore, by Lemma 3.1, we may assume that G' is reduced.

Let $A \in N'$, θ be a string over $N' \cup \Sigma \cup \{\#\}$, and $\xi = \sigma\langle \xi_1 \# \xi_2 \# \dots \# \xi_k \rangle$ in T_Σ for some $k \in \mathbb{N}$, $\sigma \in \Sigma^{(k)}$, and $\xi_1, \xi_2, \dots, \xi_k \in \mathrm{T}_\Sigma$. We claim that

$$\begin{aligned} &\text{if } A \Rightarrow_{G'} \langle\theta\rangle \Rightarrow^*_{G'} \mathcal{S}(\xi) \ , \text{ then } \theta = \sigma A_1 \# A_2 \# \dots \# A_k \\ &\text{for some } A_1, A_2, \dots, A_k \in N' \text{ with } A_i \Rightarrow^*_{G'} \mathcal{S}(\xi_i) \text{ for all } i \in [k] \ . \end{aligned} \tag{5.2}$$

Now we prove (5.2). Since G' is a parenthesis grammar, by Lemma 3.4, there do not exist $B \in N'$ and $\gamma \in \mathrm{prefix}(\sigma\mathcal{S}(\xi_1)\#\mathcal{S}(\xi_2)\# \dots \#\mathcal{S}(\xi_k))$ such that $B \Rightarrow^*_{G'} \gamma$, and thus, $\theta = \sigma\theta'$ for some string θ' over $N' \cup \Sigma \cup \{\#\}$. We proceed by case analysis.

Assume that $k = 0$. Then $\sigma = \alpha$ and $\xi = \alpha\langle\rangle$ for some $\alpha \in \Sigma^{(0)}$, and hence, $\mathcal{S}(\alpha\langle\rangle) = \langle\alpha\rangle$. Furthermore, since G' is a parenthesis grammar, we have $\theta = \alpha$ and $\theta' = \varepsilon$.

Now assume that $k > 0$. Then, since $\langle$ is in $\mathrm{prefix}(\mathcal{S}(\xi_1)\#\mathcal{S}(\xi_2)\# \dots \#\mathcal{S}(\xi_k))$ and G' is a parenthesis grammar, we must have $\theta' = A_1\theta''$ for some $A_1 \in N'$ and string θ'' over $N' \cup \Sigma \cup \{\#\}$. Since G' is a parenthesis grammar, by Lemma 3.4, for all $w \in (\Sigma^{\Xi})^*$, if $A_1 \Rightarrow^*_{G'} w$, then $w = \langle u \rangle$ for some $u \in (\Sigma^{\Xi})^*$ such that u is balanced. The one and only way to satisfy the aforementioned requirement on A_1 with respect to $A \Rightarrow_{G'} \langle\sigma A_1\theta''\rangle \Rightarrow^*_{G'} \mathcal{S}(\xi)$ is that if $A_1 \Rightarrow^*_{G'} \mathcal{S}(\xi_1)$. (Observe that, since G' is a parenthesis grammar, we have $A_1 \Rightarrow_{G'} \langle\theta_1\rangle \Rightarrow^*_{G'} \mathcal{S}(\xi_1)$ for some string θ_1 over $N' \cup \Sigma \cup \{\#\}$, which satisfies the condition of (5.2) as well.) Then, since G' is a parenthesis grammar, by Lemma 3.4, there do not exist $C \in N'$ and $v \in \mathrm{prefix}(\#\mathcal{S}(\xi_2)\# \dots \#\mathcal{S}(\xi_k))$ such that $C \Rightarrow^*_{G'} v$, and hence, $\theta'' = \#\hat{\theta}$ for some string $\hat{\theta}$ over $N' \cup \Sigma \cup \{\#\}$. Putting these together, we currently have $\theta = \langle\sigma A_1 \# \hat{\theta}\rangle$. Clearly, by continuing our argumentation in a similar way, we can show that $\theta = \sigma A_1 \# A_2 \# \dots \# A_k$ and that $A_i \Rightarrow^*_{G'} \mathcal{S}(\xi_i)$ for all $i \in [k]$. This completes the proof of (5.2).

It follows from (5.2) that each rule in R' has the form $A \to \langle\sigma A_1 \# A_2 \# \dots \# A_k\rangle$ with $k \in \mathbb{N}$, $\sigma \in \Sigma^{(k)}$, and $A, A_1, A_2, \dots, A_k \in N'$.

Next we can effectively construct the Σ-rtg $\mathcal{G} = (N', S', R'')$ such that $A \to \sigma\langle A_1 \# A_2 \# \dots \# A_k\rangle$ is in R'' iff $A \to \langle\sigma A_1 \# A_2 \# \dots \# A_k\rangle$ is in R'.

We claim that, for all $A \in N'$ and $\xi \in \mathrm{T}_\Sigma$, we have

$$A \Rightarrow^*_{G'} \mathcal{S}(\xi) \quad \text{iff} \quad A \Rightarrow^*_{\mathcal{G}} \xi \ . \tag{5.3}$$

Next we prove (5.3) by structural induction on ξ. Let $\xi = \alpha\langle\rangle$ for some $\alpha \in \Sigma^{(0)}$. Clearly, we have $\mathcal{S}(\alpha\langle\rangle) = \langle\alpha\rangle$. Moreover, we have

$$A \Rightarrow^*_{G'} \langle\alpha\rangle \quad \text{iff} \quad A \to \langle\alpha\rangle \text{ is in } R' \quad \text{iff} \quad A \to \alpha\langle\rangle \text{ is in } R'' \quad \text{iff} \quad A \Rightarrow^*_{\mathcal{G}} \alpha\langle\rangle \ .$$

Now let $\xi = \sigma\langle\xi_1 \# \xi_2 \# \dots \# \xi_k\rangle$ with $k \in \mathbb{N}_+$, $\sigma \in \Sigma^{(k)}$, and $\xi_1, \xi_2, \dots, \xi_k \in \mathrm{T}_\Sigma$. For every $A_1, A_2, \dots, A_k \in N'$, the rule $A \to \langle\sigma A_1 \# A_2 \# \dots \# A_k\rangle$ exists in R' iff the rule $A \to \sigma\langle A_1 \# A_2 \# \dots \# A_k\rangle$ exists is in R''. Moreover, by I. H., for each $i \in [k]$, we have $A_i \Rightarrow^*_{G'} \mathcal{S}(\xi_i)$ iff $A_i \Rightarrow^*_{\mathcal{G}} \xi_i$. So, we have

$$A \Rightarrow_{G'} \langle\sigma A_1 \# A_2 \# \dots \# A_k\rangle \Rightarrow^*_{G'} \langle\sigma\mathcal{S}(\xi_1) \# A_2 \# \dots \# A_k\rangle$$

$$\Rightarrow^*_{G'} \langle \sigma\mathcal{S}(\xi_1)\#\mathcal{S}(\xi_2)\#\ldots\#\mathcal{S}(\xi_k)\rangle = \mathcal{S}(\xi)$$

if and only if

$$\begin{aligned} A &\Rightarrow_{\mathcal{G}} \sigma\langle A_1\#A_2\#\ldots\#A_k\rangle \Rightarrow^*_{\mathcal{G}} \sigma\langle \xi_1\#A_2\#\ldots\#A_k\rangle \\ &\Rightarrow^*_{\mathcal{G}} \sigma\langle \xi_1\#\xi_2\#\ldots\#\xi_k\rangle = \xi \ . \end{aligned}$$

Therefore, for each $\xi \in \mathrm{T}_\Sigma$, we have

$$\mathcal{S}(\xi) \in \mathrm{L}(G') \text{ iff } S' \Rightarrow^*_{G'} \mathcal{S}(\xi) \text{ iff}^{(*)} S' \Rightarrow^*_{\mathcal{G}} \xi \text{ iff } \xi \in \mathrm{L}(\mathcal{G}) \ ,$$

where at $(*)$ we used the fact that $S' \Rightarrow^*_{G'} \mathcal{S}(\xi)$ iff $S' \Rightarrow^*_{\mathcal{G}} \xi$ by (5.3). □

Acknowledgements. The author would like to thank Joost Engelfriet for highlighting the close connection between parenthesis languages and tree languages and for pointing out the paper [8], and the anonymous referees of this paper for their work and useful suggestions.

References

[1] J. Berstel, L. Boasson: *Formal properties of XML grammars and languages*, Acta Inform. 38 (2002), pp. 649–671, doi: https://doi.org/10.1007/s00236-002-0085-4.

[2] W. S. Brainerd: *Tree Generating Regular Systems*, Inf. Control 14.2 (1969), pp. 217–231, doi: https://doi.org/10.1016/S0019-9958(69)90065-5.

[3] N. Chomsky: *Three models for the description of language*, IEEE Trans. Inf. Theory 2.3 (1956), pp. 113–124, doi: https://doi.org/10.1109/TIT.1956.1056813.

[4] J. Engelfriet: *Tree automata and tree grammars*, tech. rep. DAIMI FN-10, see also: arXiv:1510.02036v1 [cs.FL] 7 Oct 2015, Inst. of Mathematics, University of Aarhus, Department of Computer Science, Denmark, 1975.

[5] F. Gécseg, M. Steinby: *Tree Automata*, see also: arXiv:1509.06233v1 [cs.FL] 21 Sep 2015, Akadémiai Kiadó, Budapest, 1984.

[6] M. Harrison: *Introduction to Formal Language Theory*, Addison-Wesley, 1978.

[7] J. Hopcroft, J. Ullman: *Introduction to automata theory, languages, and computation*, Addison-Wesley, 1979.

[8] D. E. Knuth: *A Characterization of Parenthesis Languages*, Inf. Control 11.3 (1967), pp. 269–289, doi: https://doi.org/10.1016/S0019-9958(67)90564-5.

[9] R. McNaughton: *Parenthesis Grammars*, J. Assoc. Computing Mach. 14 (1967), pp. 490–500.

[10] J. Mezei, J. B. Wright: *Algebraic automata and context-free sets*, Inf. Control 11.1-2 (1967), pp. 3–29, doi: https://doi.org/10.1016/S0019-9958(67)90353-1.

[11] W. C. Rounds: *Tree-oriented proofs of some theorems on context-free and indexed languages*, in: Proceedings of the second annual ACM symposium on Theory of computing, 1970, pp. 109–116, doi: https://doi.org/10.1145/800161.805156.

[12] J. Thatcher: *Tree automata: an informal survey*, in: Currents in the Theory of Computing, ed. by A. V. Aho, Englewood Cliffs, N. J.: Prentice-Hall, 1973, pp. 143–172.

Annales Mathematicae et Informaticae
56 (2022) pp. 71–83
DOI: https://doi.org/10.33039/ami.2022.12.004
URL: https://ami.uni-eszterhazy.hu

Stability condition of multiclass classical retrials: a revised regenerative proof*

Evsey Morozov[abc], **Stepan Rogozin**[ab]

[a]Institute of Applied Mathematical Research, Karelian Research Centre RAS, Petrozavodsk, Russian Federation

[b]Institute of Mathematics and Information Technologies, Petrozavodsk State University, Petrozavodsk, Russian Federation

[c]Moscow Center for Fundamental and Applied Mathematics, Moscow State University, Moscow 119991, Russian Federation
emorozov@karelia.ru
ppexa@mail.ru

Abstract. We consider a multiclass retrial system with classical retrials, and present a new short proof of the sufficient stability (positive recurrence) condition of the system. The proof is based on the analysis of the departures from the system and a balance equation between the arrived and departed work. Moreover, we apply the asymptotic results from the theory of renewal and regenerative processes. This analysis is then extended to the system with the outgoing calls. A few numerical examples illustrate theoretical analysis.

Keywords: multiclass system, classical retrials, outgoing calls, stability analysis, regeneration, simulation

1. Introduction

The importance of the retrial queues to model the modern wireless telecommunication systems is well-known, for instance, see [2–4, 8], where also a comprehensive bibliography on research related to retrial queues can be found. In this work we focus on the stability analysis of a classical retrial queue, and using regenerative arguments present a new short proof of the known sufficient stability condition of

*This research was supported by the Ministry of Education and Science of the Russian Federation as part of the program of the Moscow Center for Fundamental and Applied Mathematics under the agreement № 075-15-2022-284.

Submitted: October 22, 2022
Accepted: December 3, 2022
Published online: December 28, 2022

such a system. This proof is not only much shorter than that have been used earlier (see [1, 9]) but also allows easily to cover more general retrial systems with the so-called 'outgoing' calls [11]. Although there are many papers which investigate the steady-state performance of the retrial queues, still a little attention is devoted to stability analysis outside the Markovian setting, which is the topic of this research.

In this regard, we first mention a fundamental work [1] in which a detailed stability analysis of a general $G/G/1$-type single-server retrial queue is developed. The authors study the system with a stationary input process and non-exponential retrial times, and also investigate the convergence rate to stationarity, but they do not appeal to the regenerative method. The stability of multiserver retrial systems is studied in a few papers. For instance, the paper [7] studies such a system with a finite buffer, batch Markovian arrival process, phase-type service time distribution and a general retrial rate, and stability analysis is based on the corresponding embedded Markov chains.

In the present paper, we consider the stability of a multiclass retrial $M/G/1$ queue with independent Poisson inputs of primary customers belonging to N different classes. Then we outline how this analysis is extended to a multiserver system. If an arriving class-i primary customer finds server busy, he joins the corresponding (infinite capacity) orbit i, and, after exponential time, attempts to capture server again. These attempts continue until he finds server idle. Service time as well as the retrial times are assumed to be class-dependent.

We use the regenerative approach [5, 10, 16, 17] to reprove the known stability condition, and this work thus complements our previous works [9] and [11]. In [9], the proof is based on the negative drift of the remaining work in the (single) orbit, while in the paper [11], we utilize the positive drift of the idle time of server. In a contrast, in the current analysis we observe the system at the departure instants and evaluate the idle time of servers after each departure. Then, assuming instability, the idle times decreases, and this effect, in the limit, contradicts a predefined condition. Then, according to the approach developed in [14, 15], we appeal to a characterization of the remaining regeneration time of a basic process to deduce that it is *positive recurrent*. This approach leads to a radical simplification of the stability analysis and also is extended to the system in which there are 'outgoing calls' during idle periods of the servers. The idea to consider the output process is not new of course. For instance, the analysis of $M/G/1$-type retrial system in [8] is based on the analysis of an embedded Markov chain representing the orbit size at the service completion epochs. The main feature of a retrial system from the point-of-view of stability analysis is that, in such a system, after each service completion, the server becomes idle for a random time until the beginning of the next service. This implies a loss of the server capacity after each departure, and thus the service discipline turns out to be not *work-conserving*. Fortunately, the service discipline in the retrial queueing system with classical retrials approaches the work-conserving discipline in the corresponding buffered system as the orbit size increases, and by this reason, it is *asymptotically work-conserving* [9]. This leads to the coincidence of the stability conditions in the retrial system and in the

corresponding classical buffered system.

The stability analysis then is extended to the system in which the idle server initiates an *outgoing call* [11]. Although these calls are expected to increase the utilization of the server, while keeping the throughput of the primary external customers, it is intuitive that the stability condition remains the same in this case as well, and we show it below.

The main contribution of this paper is to present a new short proof of the sufficient stability condition of this model, which then allows to study analogously the system with the outgoing calls (We note that the main result of this paper has been announced in [12].) Moreover, the approach used in this paper has a promising potential to analyse stability condition of a multiserver multicalss system in which the service times are both *class- and server-dependent*. In particular, it is demonstrated in a recent paper [13] where the stability analysis of a retrial system (a modified Erlang system) with two classes of customers and c identical servers has been performed by means of the same approach.

The rest of this paper is organized as follows. Section 2 describes the model and the regenerative structure of a basic stochastic process. In Section 3, we give the main balance equation and present the new proof of the stability condition. In Section 4, the stability analysis is extended to the system with the outgoing calls. To illustrate the theoretical results, some numerical results based on stochastic simulation are included in Section 5.

2. Description of the model

We consider a multiclass retrial $M/G/1$ queueing system with N independent Poisson inputs of primary customers, and for each class i, denote by τ_i a generic (exponential) interarrival time, with rate $\lambda_i = 1/\mathsf{E}\tau_i$, by $\{S_n^{(i)}\}$ the iid service time (with generic time $S^{(i)}$ with rate $\mu_i = 1/\mathsf{E}S^{(i)}$), and by γ_i the rate of (exponential) class-i retrial time, $i = 1, \ldots, N$. If a class-i primary customer finds server busy, he joins the corresponding (infinite capacity) orbit i and, after the exponential time with rate γ_i, the customer attempts to capture server. He continues his attempts until finds the server idle. Denote by $\gamma_0 = \min \gamma_i$. This rule is called 'classical retrial policy', and if the orbit size equals N, then the retrial rate at this instant is lower bounded by $\gamma_0 N$ by the memoryless property of the exponential distribution.

To describe the regenerative structure of the system, we denote by $Q(t)$ the total number of customers in the system at instant t^-, let $\{t_k\}$ be the arrival instants of the superposed input (Poisson) process and $Q(t_k) =: Q_k$. Then the regeneration instants $\{T_n\}$ of the process $\{Q(t),\, t \geq 0\}$ are recursively defined as

$$T_{n+1} = \inf_k (t_k > T_n : Q_k = 0),\ n \geq 0, \tag{2.1}$$

($T_0 := 0$), and the regeneration instants of the embedded process $\{Q_n\}$ are

$$\theta_{n+1} = \inf(k > \theta_n : Q_k = 0),\ n \geq 0\ \ (\theta_0 := 0). \tag{2.2}$$

The generic regeneration period, that is the distance between two arbitrary adjacent regeneration points, is denoted by T (for continuous-time construction (2.1)) and by θ (for discrete-time construction (2.2)). If the mean $\mathsf{E}T < \infty$ then the regenerative process $\{Q(t)\}$ (and the queueing system) is called *positive recurrent* (stable), and it implies the existence of the stationary process $Q(t) \Rightarrow Q$, $t \to \infty$ ($\Rightarrow$ denotes convergence in distribution). If $\mathsf{E}T = \infty$ then the system is called *null-recurrent* or unstable. By the (stochastic) equality $T =_{st} \tau_1 + \cdots + \tau_\theta$, it follows by the Wald's identity that $\mathsf{E}T = \mathsf{E}\theta\, \mathsf{E}\tau$, where both sides of the equality are finite/infinite simultaneously [6], implying that both processes $\{Q(t)\}$ and $\{Q_n\}$ are positive recurrent/null-recurrent simultaneously.

3. Stability analysis

Denote by

$$\rho_i = \lambda_i/\mu_i,\ \rho = \sum_{i=1}^{N} \rho_i. \tag{3.1}$$

Below we present a new and short proof of the following statement.

Theorem 3.1. *If $\rho < 1$ then the (initially idle) system under consideration is positive recurrent, $\mathsf{E}T < \infty$.*

Proof. Let $\{d_k,\ k \geq 1\}$ be the departure instances of the served customers leaving the system. Denote by $V_i(t)$ the total workload which class-i customers bring in the system in time interval $[0,\ t]$ and let $V(t) = \sum_i V_i(t)$. Moreover denote by $W(t)$ the remaining work in all orbits at instant t, and let $I(t)$ be the total idle time of the server in interval $[0,\ t]$. (All processes we consider are right-continuous with left-hand limits [15].) Finally, denote by

$$W(d_n) = W_n,\ V(d_n) = V_n,\ I(d_n) = I_n,\ n \geq 1.$$

By assumption, the first customer arrives at instant $t_1 = 0$ in the empty system (it is called *zero initial state*), and we obtain the following balance equation

$$W_n = V_n - d_n + I_n,\ n \geq 1. \tag{3.2}$$

We notice that the arrived in the interval $[0,\ d_n]$ class-i work can be written as

$$V_i(d_n) = \sum_{k=1}^{A_i(d_n)} S_k^{(i)},$$

where $A_i(d_n)$ is the number of class-i arrivals in $[0,\ d_n]$, $i = 1, \ldots, N$, $n \geq 1$. It is easy to find, using the Strong Law of Large Numbers and the property of the *cumulative processes* [17] that with probability 1 (w.p.1)

$$\lim_{n\to\infty} \frac{V_i(d_n)}{d_n} = \rho_i,$$

regardless of whether the system is stable or not, implying

$$\lim_{n\to\infty} \frac{V_n}{d_n} = \rho. \tag{3.3}$$

(Indeed it follows from theory of cumulative processes, that convergence in mean in (3.3) holds as well.) Now we assume, by a contradiction, that the system is null-recurrent that is $\mathsf{E}\theta = \infty$. (Then $\mathsf{E}T = \infty$ as well, see [6].) Note that θ is a generic number of arrivals and departures within a regeneration period of the system. By a characterization of the renewal process [15], it then follows from $\mathsf{E}\theta = \infty$ that the remaining regeneration time

$$\theta(n) := \inf_k(\theta_k - n : \theta_k - n > 0), \tag{3.4}$$

at instant n up to the next regeneration instant, increases to infinity *in probability*, that is

$$\theta(n) \Rightarrow \infty,\ n \to \infty. \tag{3.5}$$

Denote $Q(d_n) = Q_n$. Using a proof by contradiction we can show (see [15]) that $Q_n \not\Rightarrow \infty$ implies $\mathsf{E}\theta < \infty$. Thus it follows from (3.5) that $\mathsf{E}\theta = \infty$, and then we obtain as well

$$Q_n \Rightarrow \infty,\ \ n \to \infty. \tag{3.6}$$

Denote by $\Delta_k = I(d_{k+1}) - I(d_k)$ the idle time of server between the kth and $(k+1)$th departures. We note that, provided $Q_k \geq n$, the mean idle time of server after the kth departure is upper bounded by the constant

$$C_n := 1/(\lambda + n\gamma_0), \tag{3.7}$$

and $C_n \to 0$ as $n \to \infty$. (Recall that γ_0 is the minimal retrial rate.) This shows that, if $Q_k \geq n$, then $\mathsf{E}\Delta_k \leq C_n$ can be done arbitrarily small for n large enough $(n \geq k)$. Then one can show that for an arbitrary $\varepsilon > 0$

$$\mathsf{E}I_n \leq \varepsilon n(1 + 1/\lambda) + L,$$

where L is a constant. (For details see formulas (21)-(25) in the paper [13].) This implies that, under assumption (3.6),

$$\lim_{n\to\infty} \frac{\mathsf{E}I_n}{n} = 0. \tag{3.8}$$

We assume that if the nth customer entering server belongs to class i, then we assign the service time $S_n^{(i)}$ from the corresponding iid sequence $\{S_n^{(i)}\}$ initially intended for this class of customers. (In other words, we omit not used elements of this sequence.) Now we consider the 'minimal' service times realized by the server,

$$S_n^{(0)} = \min_{1\leq i\leq N} S_n^{(i)},\ n \geq 1.$$

These times constitute an iid sequence $\{S_n^{(0)}\}$ with generic element $S^{(0)}$. Now we define a random walk

$$\widehat{d}_n = \sum_{k=1}^{n} S_k^{(0)}, \ \ n \geq 1. \tag{3.9}$$

An important observation is that $\widehat{d}_n \leq d_n$, $n \geq 1$. Now we can write

$$\frac{I_n}{d_n} \leq \frac{I_n}{\widehat{d}_n} = \frac{I_n}{n}\frac{n}{\widehat{d}_n}, \ \ n \geq 1.$$

On the other hand, we have, from the renewal theory, that w.p.1,

$$\lim_{n\to\infty} \frac{n}{\widehat{d}_n} = \frac{1}{\mathsf{E}S^{(0)}}. \tag{3.10}$$

Then it follows from (3.8) that, as $n \to \infty$,

$$\frac{I_n}{n} \Rightarrow 0.$$

In turn, then there exists a subsequence $n_k \to \infty$, $k \to \infty$, such that

$$\lim_{k\to\infty} \frac{I_{n_k}}{n_k} = 0, \tag{3.11}$$

w.p.1, see [6]. Now we return to the balance equation (3.2) written for the subsequence $\{n_k\}$:

$$W_{n_k} = V_{n_k} - d_{n_k} + I_{n_k}, \ \ n \geq 1. \tag{3.12}$$

Note that $\widehat{d}_{n_k} \to \infty$ w.p.1 and, as in (3.10),

$$\lim_{k\to\infty} \frac{n_k}{\widehat{d}_{n_k}} = \frac{1}{\mathsf{E}S^{(0)}}.$$

Thus, w.p.1, as $k \to \infty$,

$$\frac{I_{n_k}}{d_{n_k}} = \frac{I_{n_k}}{n_k}\frac{n_k}{d_{n_k}} \leq \frac{I_{n_k}}{n_k}\frac{n_k}{\widehat{d}_{n_k}} \to 0. \tag{3.13}$$

Now we divide both sides of (3.12) by d_{n_k} and let $k \to \infty$. Because

$$\lim_{k\to\infty} \frac{W_{n_k}}{d_{n_k}} \geq 0 \tag{3.14}$$

(this limit exists because the r.h.s. limit of (3.12) exists), then we obtain that

$$\rho \geq 1, \tag{3.15}$$

implying a contradiction with the assumption $\rho < 1$. In other words,

$$Q(d_n) \not\Rightarrow \infty, \ n \to \infty,$$

and there exists a subsequence $\{z_k\}$, $z_k \to \infty$ and some $\varepsilon > 0$ and constant $C < \infty$, such that

$$\inf_k \mathsf{P}(Q(d_{z_k}) \le C) \ge \varepsilon.$$

We note that a 'regeneration' condition

$$\min_{1 \le i \le N} \mathsf{P}(\tau > S^{(i)}) > 0$$

in this system holds automatically because the input process is Poisson, see [15]. Then by a standard method [15] we can show that the remaining regeneration time (3.4) (measured in the number of the arrivals/departures within a cycle)

$$\theta(n_k) \not\Rightarrow \infty, \tag{3.16}$$

which in turn implies that the mean number of arrivals/departures within a regeneration cycle $\mathsf{E}\theta < \infty$. Finally,

$$\mathsf{E}T = \mathsf{E}\theta\, \mathsf{E}\tau < \infty, \tag{3.17}$$

and the proof of Theorem 3.1 is hereby completed. □

Remark 3.2. The prove given above is radically shorter and more intuitive than that have been obtained in previous works [9, 11, 15]. This also relates to the 2nd step of the stability analysis describing the so-called 'unloading' of the system after the step (3.16).

Remark 3.3. It follows from (3.17) that, under assumption $\rho < 1$, the continuous-time processes and the embedded processes (both at the instants $\{t_n\}$ and $\{d_n\}$) are positive recurrent.

Assume that the system has $m \ge 1$ identical servers. In this system definition of the regeneration points remains the same as in (2.1). In this case, in the balance equation (3.2), d_n is replaced by md_n and the total idle time of all servers in interval $[0, d_n]$ becomes

$$I_n = \sum_{j=1}^{m} \sum_{k=1}^{n-1} \Delta_k^{(j)},$$

where $\Delta_k^{(j)}$ denotes the idle time of server j after the kth departure. The extension of stability analysis to this case is straightforward, and the proof of the following Theorem 3.4 follows mainly the same lines as the proof of Theorem 3.1.

Theorem 3.4. *If $\rho < m$ then the (initially idle) system is positive recurrent, $\mathsf{E}T < \infty$.*

4. Extensions to the system with outgoing calls

Now we consider a single-server system with *outgoing calls*. Keeping the main notation, we assume that, provided the server is idle, it generates an outgoing class-i call with rate ν_i, and there are K different types of such calls. Denote the total rate by $\nu = \sum_{i=1}^{K} \nu_i$. The service times (durations) of class-i calls are iid $\{Z_n^{(i)}, n \geq 1\}$ with the mean $\mathsf{E}Z^{(i)} < \infty$, $i = 1, \ldots, K$. (We omit serial index denoting a generic element of an iid sequence.) In this system the balance equation, again considered at the departure instants $\{d_n\}$, becomes

$$V_n + \mathsf{Z}_n = W_n + d_n - I_n, \tag{4.1}$$

where Z_n is the workload generated by the outgoing calls in the interval $[0, d_n]$. It is assumed that the service of an outgoing call is not interrupted by a newly arrived external customer. Note that in this case the upper bound C_n in (3.7) is modified as follows:

$$C_n = \frac{1}{\lambda + n\gamma_0 + \nu}.$$

Now we denote

$$S_n^{(0)} = \min_{1 \leq i \leq N} S_n^{(i)}, \qquad Z_n^{(0)} = \min_{1 \leq i \leq K} Z_n^{(i)} \ n \geq 1,$$

and redefine the instants $\widehat{d}_n$ (see (3.9)) as

$$\widehat{d}_n = \sum_{k=1}^{n} \min\{S_k^{(0)}, Z_k^{(0)}\}, \ \ n \geq 1.$$

As above, $d_n \geq \widehat{d}_n \to \infty$. Assume again that convergence (3.6) holds true. Then, as in Section 3, there exists a subsequence $d_{n_k} \to \infty$, $k \to \infty$, satisfying (3.11) (not necessary the same one). Rewrite the balance equation (4.1) as

$$V_{n_k} + \mathsf{Z}_{n_k} = W_{n_k} + d_{n_k} - I_{n_k},$$

and show that, as $k \to \infty$,

$$\frac{\mathsf{Z}_{n_k}}{d_{n_k}} \leq \frac{\mathsf{Z}_{n_k}}{\widehat{d}_{n_k}} \to 0 \quad \text{w.p.1.} \tag{4.2}$$

Denote by N_{n_k} the number of events in the Poisson process with rate ν in time interval $[0, I_{n_k}]$. Then it is easy to see that $N_{n_k} \geq_{st} \widehat{N}_{n_k}$, where $\widehat{N}_{n_k}$ is the number of actual outgoing calls generated in interval $[0, d_{n_k}]$. It is because some calls, among (maximally possible) number N_{n_k}, are 'lost' (if server transmits another outgoing call), and in result $\widehat{N}_{n_k}$ in general turns out to be less than N_{n_k}. Denote by v_n the work which call n brings in the system, $n \geq 1$. Then v_n is (stochastically) upper bounded as

$$v_n \leq_{st} Z_n^{(1)} + \cdots + Z_n^{(K)} =: \mathcal{Z}_n,$$

where iid $\{\mathcal{Z}_n\}$ have generic element $\mathcal{Z}$ with mean $\mathsf{E}\mathcal{Z} = \sum_{i=1}^{K} \mathsf{E}Z^{(i)} < \infty$. It now follows from above the following stochastic inequality

$$\mathsf{Z}_{n_k} = \sum_{j=1}^{\widehat{N}_{n_k}} v_j \leq_{st} \sum_{j=1}^{N_{n_k}} \mathcal{Z}_j. \tag{4.3}$$

We note that, on the event $\{\sup_k I_{n_k} < \infty\}$, it follows that $\sup_k \mathsf{Z}_{n_k} < \infty$ as well (because the number of the events in any finite interval is finite w.p.1), implying $\mathsf{Z}_{n_k} = o(\widehat{d}_{n_k})$, $k \to \infty$. Otherwise, on the event $\{\lim_{k\to\infty} I_{n_k} = \infty\}$, we can write, by (4.3)

$$\frac{\mathsf{Z}_{n_k}}{\widehat{d}_{n_k}} \leq \frac{1}{N_{n_k}} \sum_{j=1}^{N_{n_k}} \mathcal{Z}_j \, \frac{N_{n_k}}{I_{n_k}} \, \frac{I_{n_k}}{\widehat{d}_{n_k}},$$

and now (4.2) follows from (3.13) because, by the Strong Law of Large Numbers,

$$\lim_{k\to\infty} \frac{1}{N_{n_k}} \sum_{j=1}^{N_{n_k}} \mathcal{Z}_j = \mathsf{E}\mathcal{Z} < \infty,$$

and by the renewal theory,

$$\lim_{k\to\infty} \frac{N_{n_k}}{I_{n_k}} = \nu < \infty.$$

It now follows that (4.2) holds and, as in (3.14) (in notation (3.1)), we arrive to the contradictory condition (3.15). The above analysis can be summarized as the following statement.

Theorem 4.1. *If $\rho < 1$ then the initially idle retrial system with the outgoing calls is positive recurrent.*

5. Simulation

The purpose of the numerical result presented below (and based on the stochastic discrete-event simulation) is to demonstrate the *asymptotically work-conserving property* meaning that, as the orbit sizes increase, the dynamics of the service process becomes similar to that in the classic buffered system [9]. By this reason we consider the border of the stability region, that is $\rho = 1$, in which case the system becomes unstable. (An exception is the experiment shown on Fig. 7.) Also we analyze stability of the multiserver system (to illustrate Theorem 3.4) and we demonstrate simulation results for a three-server system. More exactly, we consider two-class retrial systems with input rates $\lambda_1 = 20/3$, $\lambda_2 = 10/3$ and with 1 and 3 servers, respectively.

To keep value $\rho = 1$, we take service rate $\mu = 10$ for 1-server system, and $\mu = 10/3$ for each server in 3-server system. Fig. 1 demonstrates a decreasing of

the expected idle periods $\mathsf{E}\Delta_k$, as k increases, for exponential (Exp) and Pareto

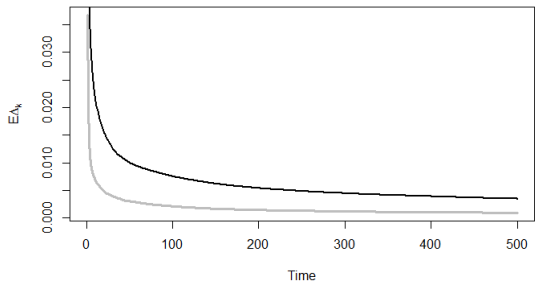


Figure 1. The expected idle time $\mathsf{E}\Delta_k$ vs. simulation time: Exp. service time (grey) and Pareto service time with $\alpha = 2$ (black); retrial rates $\gamma_1 = \gamma_2 = 30$.

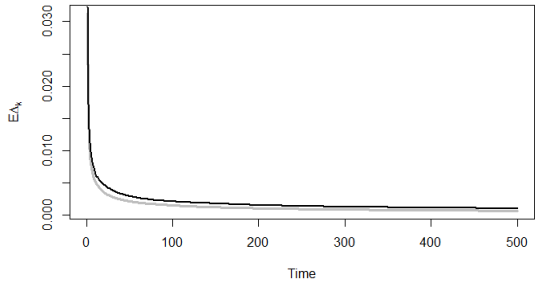


Figure 2. The expected server idle time in the system with outgoing calls vs. simulation time: Exp. service time (grey), Pareto service time (black); $\gamma_1 = \gamma_2 = 30$, $\nu = 30$.

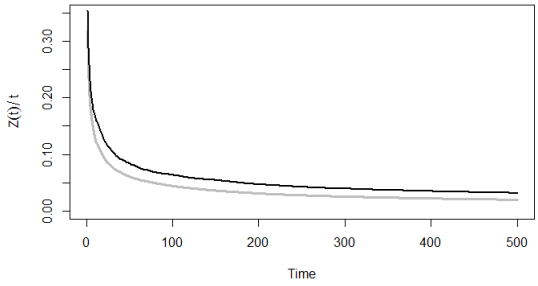


Figure 3. The workload of the outgoing calls with Weibull service time vs. modeling time: Exp. service time (grey), Pareto service time (black); $\gamma_1 = \gamma_2 = 30$, $\nu = 30$.

service time distribution

$$F(x) = 1 - \left(\frac{x_0}{x}\right)^{\alpha}, \quad x \geq x_0, \tag{5.1}$$

where parameter $x_0 = 0.05$ for 1-server system, and $x_0 = 0.15$ for 3-server system,

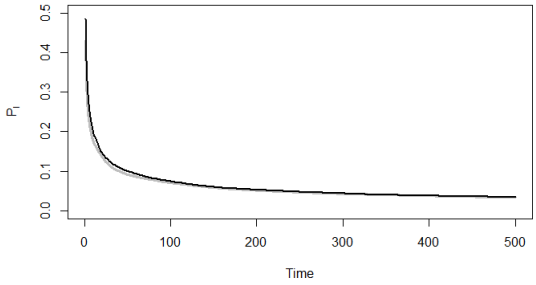


Figure 4. The idle server probability P_I in the original system vs. simulation time: 1 server (grey) and 3 servers (black); $\gamma_1 = \gamma_2 = 30$.

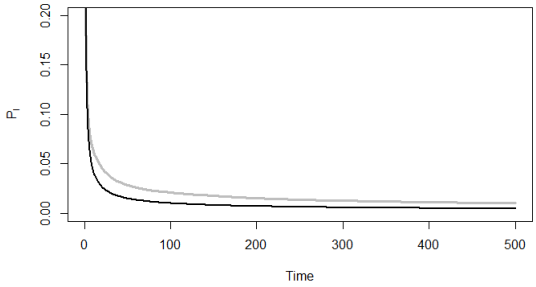


Figure 5. The idle server probability P_I vs. simulation time in the system with outgoing calls: 1 server (grey) and 3 servers (black); $\gamma_1 = \gamma_2 = 30$, $\nu = 30$.

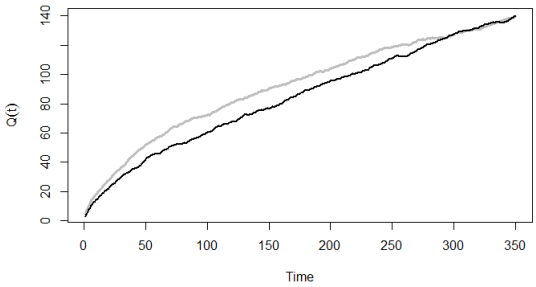


Figure 6. The orbit size in retrial system (grey) and buffer size in the buffered system (black), with Pareto service time, vs. simulation time; $\gamma_1 = \gamma_2 = 10$.

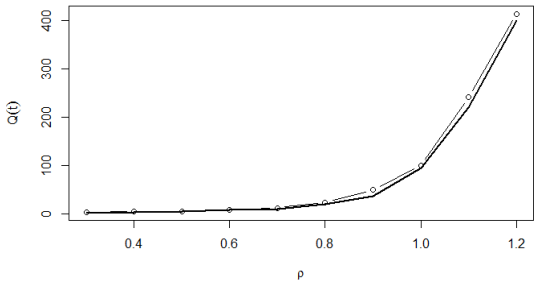


Figure 7. The orbit size in the original retrial system (dotted line) and the buffer size in the buffered system (solid line), with Pareto service time, vs. traffic intensity ρ; $\gamma_1 = \gamma_2 = 10$.

and parameter $\alpha = 2$ in all cases. Fig. 2 shows a similar result for the system with one class of the outgoing calls with rate $\nu = 30$. The service times of the outgoing calls have Weibull distribution

$$F(x) = 1 - \exp\Big\{ - \Big(\frac{x}{b}\Big)^a \Big\},\ x \geq 0,$$

with the shape parameter $a = 0.9$, while the scale parameter $b = 0.1$ for 1-server system and $b = 0.3$ for 3-server system, respectively.

Thus Fig. 1 and Fig. 2 confirm the limit (3.11) and the asymptotic work-conserving property both for the original one -server retrial system and for the corresponding system with the outgoing calls, when $\rho = 1$. Fig. 3 describes the same effect expressed as a vanishing fraction of the workload generated by the outgoing calls, if $\rho = 1$, confirming the limit (4.2). Similar results given on Fig. 4, Fig. 5 show that the idle time fraction ('idle time probability') goes to zero in all considered systems, but this convergence is faster in the system with outgoing calls. Finally, we compare the orbit size in the retrial system and the buffer size in the corresponding buffered system with Pareto service time (5.1) (see Fig. 6, Fig. 7). In particular, Fig. 6 shows that (for $\rho = 1$) the orbit size initially increases faster than the buffer size but then they behave similarly. Fig. 7 also demonstrates the proximity between the orbit size and buffer size for the different values of the traffic intensity ρ (modeling time is 200 slots). Nevertheless, the orbit size is slightly larger than the buffer size, and it is an expected result.

6. Conclusion

In this work, we develop a new and short regenerative proof of the stability condition of a multiclass retrial system with classical retrials. Unlike previous proofs, we focus on the departures of customers and, provided the number of orbital customers increases, show a contradiction with a predefined negative drift condition, which turns out to be a sufficient stability condition. This approach is then extended to

the system with outgoing calls, and it has a promising potential in the stability analysis of more general retrial systems. Some numerical examples based on the simulation are given which illustrate the asymptotically work-conserving property of the system with classical retrials.

References

[1] E. Altman, A. A. Borovkov: *On The Stability of Retrial Queues*, Queueing systems 26 (1997), pp. 343–363, doi: https://doi.org/10.1023/A:1019193527040.

[2] J. R. Artalejo: *A classified bibliography of research on retrial queues: Progress in 1990-1999*, Top 7 (1999), pp. 187–211, doi: https://doi.org/10.1007/BF02564721.

[3] J. R. Artalejo: *Accessible bibliography on retrial queues*, Mathematical and Computer Modelling 30 (1999), pp. 1–6, doi: https://doi.org/10.1016/S0895-7177(99)00128-4.

[4] J. R. Artalejo, G. I. Falin: *Standard and retrial queueing systems: A comparative analysis*, Revista Matematica Complutense 15 (2002), pp. 101–129, doi: https://doi.org/10.5209/rev_REMA.2002.v15.n1.16950.

[5] S. Asmussen: *Applied Probability and Queues*, New York, NY: Springer-Verlag, 2003, doi: https://doi.org/10.1007/b97236.

[6] A. A. Borovkov: *Probability Theory*, Springer Science and Business Media, 2013, doi: https://doi.org/10.1007/978-1-4471-5201-9.

[7] L. Breuer, A. Dudin, V. Klimenok: *A retrial BMAP/PH/N system*, Queueing Systems 40 (2002), pp. 433–457, doi: https://doi.org/10.1023/A:1015041602946.

[8] G. I. Falin, J. G. C. Templeton: *Retrial Queues*, London: Chapman and Hall, 1997.

[9] E. Morozov: *A multiserver retrial queue: Regenerative stability analysis*, Queueing Systems 56.3 (2007), pp. 157–168, doi: https://doi.org/10.1007/s11134-007-9024-y.

[10] E. Morozov, R. Delgado: *Stability analysis of regenerative queues*, Automation and Remote control 70.3 (2009), pp. 1977–1991, doi: https://doi.org/10.1134/S0005117909120066.

[11] E. Morozov, T. Phung-Duc: *Stability analysis of a multiclass retrial system with classical retrial policy*, Performance Evaluation 122 (2017), pp. 15–26, doi: https://doi.org/10.1016/j.peva.2017.03.003.

[12] E. Morozov, S. Rogozin: *Stability analysis of classical retrials: a revised regenerative proof*, in: Proceedings of the 20th International Conference Informational Technologies And Mathematical Modelling (ITMM-2021) named after A. F. Terpugov, Tomsk, Russia: Scientific Technology Publishing House, 2022, pp. 82–87.

[13] E. Morozov, S. Rogozin, H. Q. Nguyen, T. Phung-Duc: *Modified Erlang Loss System for Cognitive Wireless Networks*, Mathematics 10.12 (2017), Art. No. 2101 (20 p.) doi: https://doi.org/10.3390/math10122101.

[14] E. Morozov, B. Steyaert: *A stability analysis method of regenerative queueing systems*, in: Queueing Theory 2, Advanced Trends. V. Anisimov, N. Limnios, New York: Wiley/ISTE, 2021, doi: https://doi.org/10.1002/9781119755234.ch7.

[15] E. Morozov, B. Steyaert: *Stability Analysis of Regenerative Queueing Models*, Springer, 2021, doi: https://doi.org/10.1007/978-3-030-82438-9.

[16] K. Sigman, R. W. Wolff: *A review of regenerative processes*, SIAM Review 35 (1993), pp. 269–288, doi: https://doi.org/10.1137/1035046.

[17] W. L. Smith: *Regenerative stochastic processes*, Proceedings of Royal Society, Ser. A 232 (1955), pp. 6–31, doi: https://doi.org/10.1098/rspa.1955.0198.

Annales Mathematicae et Informaticae
56 (2022) pp. 84–94
DOI: https://doi.org/10.33039/ami.2022.12.008
URL: https://ami.uni-eszterhazy.hu

Regeneration estimation in partially stable two class retrial queue*

Ruslana Nekrasova[ab]

[a]Institute of Applied Mathematical Research, Karelian Research Centre of RAS

[b]Petrozavodsk State University
ruslana.nekrasova@mail.ru

Abstract. We consider two-class retrial queueing system with constant retrial rate fed by Poisson input and apply regenerative confidence estimation for mean number of customers in the stable orbit, while the other orbit intimately grows. The simulation results illustrate that partially stable case providing accurate confidence estimation, even the stability conditions, related for the whole system, are violated.

Keywords: retrial queue, constant retrial rate, stability, partial stability, regeneration estimation

AMS Subject Classification: 60, 62

1. Introduction

The paper deals with a single server retrial rate queuing system under constant retrial rate policy. The model admits two classes of customers, arrivals join the system according to Poisson input. The service times are independent and identically distributed among the corresponding class. If the server is busy at arrival instant, the new customer joins the orbit associated with its class and then try to occupy the server after class-dependent exponentially distributed retrial time according to FIFO discipline.

Retrial systems have a huge sphere of modern applications. For instance, such models successfully describe various call centers [16, 19] or the work of computer networks and internet protocols [7, 8]. The applications of retrial models to wireless

*The publication has been prepared with the support of Russian Science Foundation according to the research project No.21-71-10135, https://rscf.ru/en/project/21-71-10135/.

Submitted: October 21, 2022
Accepted: December 3, 2022
Published online: December 28, 2022

technologies are presented in [9, 15]. Retrial queuing systems are widely studied in literature, it is worth mentioning the basic books and surveys [1, 2, 10, 18].

We consider two class retrial system in partially stable mode: the first class orbit is stochastically bounded and the second class orbit infinitely grows in probability. To define partially stability conditions we rely on the preliminary results obtained in [6] and developed in [5]. The basic goal of the paper is to construct confidence interval for mean number of customers in the first orbit in case of partially stable mode. We apply regenerative method of confidence estimation. Generally, the regenerative method is applicable if the system under consideration is stable. The novelty of the present research is the following: we use regenerative approach to obtain confidence interval in case the only orbit is stochastically bounded while stability conditions for the whole system are violated.

The paper is organized as follow. Section 2 contains the detailed description of the system under consideration. Section 3 presents the concept of partial stability and known conditions for the partially stable mode. Next in Section 4 we briefly discuss the regenerative method of confidence estimation. Section 5 containes simulation results for partially stable model. We compare obtained confidence intervals with the results for corresponding single orbit retrial system in a stable mode. Section 6 concludes the paper.

2. Description of the model

We consider a single-server bufferless retrial system under *constant retrial rate* policy denoted by system Σ. The model admits two classes of customers. Namely arrivals form the superposition of two Poisson inputs with corresponding rates λ_i, where $i = 1, 2$ defines the class number. Thus the total input rate is the following: $\lambda = \lambda_1 + \lambda_2$. We define the sequence of arrival instants by $\{t_n,\, n \geq 1\}$. Note that interarrival times $\tau_n = t_{n+1} - t_n$ are independent and exponentially distributed with a rate λ. Let τ define the generic interarrival time, thus $\mathsf{E}\tau = 1/\lambda$.

Next we assume that class-i service times are independent, generally distributed and stochastically equivalent to $S^{(i)}$ with corresponding mean $1/\mu_i$. Define the marginal load coefficient by

$$\rho_i = \lambda_i/\mu_i.$$

Thus the total load coefficient is obtained as

$$\rho = \rho_1 + \rho_2.$$

If the class-i arrival, that meets the server busy, joins the corresponding infinite-capacity virtual *orbit* and then tries to occupy the server after an exponential time with a rate α_i. We define the auxiliary load coefficient associated with class-i orbit customers by

$$\hat{\rho}_i = \alpha_i/\mu_i.$$

The total orbits load coefficient is the following

$$\hat{\rho} = \hat{\rho}_1 + \hat{\rho}_2.$$

Consider $N^{(i)}(t)$ – the number of customers at orbit i at time instant t. The total number of customers in the system Σ is defined by the following process

$$X(t) = \nu(t) + N^{(1)}(t) + N^{(2)}(t), \quad t \geq 0, \tag{2.1}$$

where $\nu(t) \in \{0, 1\}$ represents the number of customers on service. Thus the only reason for unstable behavior of the system is the infinite growth of orbits size. (Note that the term "size" actually means the number of customers on the orbit, while the configuration of the system admits the infinite number of waiting places for orbit calls).

Constant retrial rate policy implies that the orbit rates α_i are fixed and do not depend on the processes $N^{(i)}(t)$. Unlike the classical retrial models, where the intensity of orbit customers increases proportionally to its number. Thus in classical multi-orbit case, the behavior of one (at instance, class-i_0) orbit affects to other orbit(s). Namely when the load of class-i_0 customers increases, the corresponding orbit size grows, and the server attack in more intensive. This implies more load to the other orbits and the growth of their sizes. Thus in classical retrial models instability of one orbit leads to the instability of other orbits. Such a property does not hold for constant retrial rate model, considered in present paper: the orbit size does not affect the intensity of orbit customers, and one orbit can infinitely grow, while the other is stable. In such a case the phenomenon of *partial stability* arises.

3. Partial stability: preliminary results

In this section we refer to the known results related to the conditions of partially stable regime in two-class retrial model with constant retrial rate. First we briefly discuss the stability concept. Note that all considered continuous-time processes are assumed to be defined at instant t^-. Each instant t_n when the new arrival joins into totally empty system ($X(t_n) = 0$) the model starts over in stochastic sense or *regenerates*. From this point of view the process $X(t)$ is called a *regenerative process*. The regenerative process is called *positive recurrent* if regeneration period has finite mean. In zero-delayed case positive recurrence implies that the system possesses have *stationary regime* [3]. Actually the positive recurrence of the process X means that starting from the arbitrary instant t the system becomes empty in a finite time. In this case we define that the system is *stable*. From this point of view the stability is equivalent to the positive recurrence. Detailed description of the regeneration approach to the stability analysis could be found in [12–14, 17].

By *partial stability* (of class-1 orbit) we define the case when class-1 orbit size process stays tight while class-2 orbit increases unlimited in probability. Note the process $N^{(1)}$ is tight [17] if for any $\delta > 0$ exists a finite constant $C \geq 0$ such that

$$\inf_t \mathsf{P}\big(\mathsf{N}^{(1)}(\mathsf{t}) \leq \mathsf{C}\big) \geq 1 - \delta. \tag{3.1}$$

Namely we obtain that only the first orbit is stochastically bounded. (Obviously that the symmetric case for partial stability of class-2 orbit is defined in analogical terms.)

Consider the absolutely continuous distribution function F with density f, defined for all x such that $1 - F(x) > 0$. Next define the failure rate by $r(x) := f(x)/\big(1 - F(x)\big)$. We say that the distribution F belongs to a special sub-class $\mathcal{D}$ if $\inf_{x\geq 0} r(x) > 0$.

The conditions of partially stable regime were firstly formulated in [6] for the multi-class retrial model, where service time distributions belongs The partial stability conditions for the model Σ considered in present paper (when class-1 orbit is tight) was obtained in [5] via load coefficients as follows:

$$\hat{\rho}_1 > \rho_1(\rho + \hat{\rho}), \tag{3.2}$$

$$\rho > \hat{\rho}_2/(\rho_2 + \hat{\rho}_2). \tag{3.3}$$

Note that to obtained the conditions (3.2), (3.3) the authors in [5] had analyzed two-dimensional Markov Chain

$$\mathbf{Y} = \big\{Y_k^{(1)}, Y_k^{(2)}\big\}, \quad k \geq 1,$$

associated with corresponding numbers of customers in the first and in the second orbit just after the departure instants (k defines the actual number of departures from the system after its service completion). The Markov property holds for the random sequences $\{Y_k^{(i)}, k \geq 1\}$, $i = 1, 2$ because input stream is assumed to be Poisson.

Relying on the technique presented in [11], it is possible to show that under conditions (3.2), (3.3) the Markov Chain $\mathbf{Y}$ is *transient*. Such a transient case is illustrated by the stability of the first orbit dynamics and the infinite growth of the second one, see [5] for details. Moreover under assumption that service time distributions belong to the sub-class $\mathcal{D}$ the conditions (3.2), (3.3) coincide with the partial stability conditions from [6]. Note that positive recurrence of $\mathbf{Y}$ implies the stability for the model Σ and corresponds to the positive recurrence of the basic process X.

Next our goal is to explore the behavior of the model under consideration when (3.2), (3.3) hold true. In this case we can expect that after some finite instant the second orbit is not empty and the total load to the server is equivalent to the load in the single-orbit retrial system, where class-2 customers arrive with a rate $\lambda_2 + \alpha_2$ and are lost in case the server is busy at arrival instants. Then we construct the auxiliary process in original two-orbit system Σ as follows:

$$X^{(1)}(t) = \nu(t) + N^{(1)}(t), \quad t \geq 0$$

and its discrete analogue $X_n^{(1)} = X^{(1)}(t_n^-)$, $n \geq 1$. Next consider the sequence

$$\beta_k = \min_n\{n > \beta_{k-1} : X_n^{(1)} = 0\}, \quad k \geq 1, \beta_0 = 0,$$

which defines the numbers of arrivals to the system when the server is idle and the first orbit is empty. Thus $\{\beta_k\}$ represents the regeneration points of the process

$\{X_n^{(1)}\}$. Next we define the sequence of independent and identically distributed (iid) regeneration cycles length in discrete time (with a generic length B) by

$$B_k = \beta_k - \beta_{k-1}, \quad n \geq 1.$$

Note that under conditions (3.2), (3.3) the process $\{X(t)\}$ (and the whole system) does not regenerate at instants t_{β_k}, while the process $X^{(1)}$ is positive recurrent. Partial stability conditions consider solely the tightness of the first orbit size process (3.1), which allows to show that with a positive probability the process $X^{(1)}$ reaches the zero value in a finite time, hence $\mathsf{EB} < \infty$.

In case the positive recurrence we can apply regeneration method (RM) for the system under consideration. RM is a powerful tool in stochastic analysis, in the next section rely on the regeneration confidence estimation to bound the dynamics of the first orbit size in partially stable regime.

4. Regenerative estimation

Recall the regenerative process $X_n^{(1)}$, which is the positive recurrent under conditions (3.2), (3.3). Note that in this case the orbit size process $N_n^{(1)}$ also regenerates with regeneration points $\{\beta_k\}$. In present section we construct the interval estimators for the mean value of the process $N_n^{(1)}$. Consider iid accumulated numbers of customers in the first orbit over the k-th regeneration cycle by

$$Z_k = \sum_{j=\beta_{(k-1)}}^{(\beta_k)-1} N_j^{(1)}, \quad k \geq 1.$$

By the results from regeneration theory and in case of positive recurrence, the following limit exists:

$$r_k := \frac{\sum_{j=1}^k Z_j}{\sum_{j=1}^k B_j} \to \frac{\mathsf{EZ}}{\mathsf{EB}} =: r, \qquad k \to \infty, \tag{4.1}$$

where Z is a generic element of a sequence $\{Z_k, k \geq 1\}$.

Note, that r_k coincides with an average number of customers in the first orbit within interval $[0, t_{\beta_k})$:

$$r_k = \frac{1}{\beta_k} \sum_{j=1}^{\beta_k} N_j^{(1)}.$$

Actually, the result (4.1) means that with a growth of cycle number, time average value of regenerative process converges to the ratio of mean cumulative value over cycle to mean cycle length. Namely, in case of positive recurrence, the behavior of regenerative process could is described by its cycle characteristics.

By Proposition 4.1 from [4] the estimator r_k satisfies the following Central Limit Theorem

$$\sqrt{k}\big(r_k - r\big) \Rightarrow \mathbb{N}(0,\, \sigma^2), \qquad n \to \infty, \tag{4.2}$$

where

$$\sigma^2 = \frac{E[Z - rB]^2}{\big(EB\big)^2}$$

and $\mathbb{N}(0,\, \sigma^2)$ is a normal distribution with zero mean. Hence, if limit (4.1) exists, then weak convergence (4.2) holds and implies the following $100(1-\gamma)\%$ confidence interval:

$$r \in \Big[r_k - \Delta_k,\, r_k + \Delta_k\Big], \tag{4.3}$$

with the accuracy

$$\Delta_k = \frac{z_\gamma \overline{\sigma}_k}{\sqrt{k}}.$$

Note, that γ is a given reliability and

$$\overline{\sigma}_k^2 = \frac{k^2}{k-1} \frac{\sum_{i=1}^k \big(Z_i - r_k B_i\big)^2}{\big(\sum_{i=1}^k B_i\big)^2}.$$

(The value z_γ defines $(1-\gamma/2)$-quantile of the standard normal law.)

4.1. Single-orbit system

The sequence $\{\beta_k\}$ does not detect regenerations of the whole sequence Σ, and we analyze the positive recurrent process $X^{(1)}$ to obtain the confidence interval (4.3).

Next we construct an additional single orbit retrial model denoted by $\hat{\Sigma}$ as follows: the input stream is fed by Poisson process with a total rate $\lambda_1 + \lambda_2 + \alpha_2$, the new arrival belongs to class 1 with a probability

$$\frac{\lambda_1}{\lambda_1 + \lambda_2 + \alpha_2}.$$

Service times are iid, class-dependent and stochastically equivalent to the corresponding service times $\{S_n^{(i)}\}$ previously defined for a model Σ. If class-1 arrival met the busy server it joins the orbit with a constant retrial rate α_1, while the second class arrival in this case leaves the system. We can expect that such new system is not less loaded than original system Σ: in case the second orbit is empty, the server is attacked by the Poisson input with a rate $\lambda_1 + \lambda_2$ and the customers from the orbit 1 (if any). In case $N_n^{(2)} > 0$ both models Σ and $\hat{\Sigma}$ behave equivalently in the sense of server load. The convergence of the first orbit size process in Σ to the orbit size in $\hat{\Sigma}$ is illustrated in [5].

The model $\hat{\Sigma}$ strictly regenerates when arrivals join into totally empty system. Denote the generic regeneration cycle length by $\hat{B}$. Stability condition for such a model is defined as follows, see [6]:

$$\hat{\rho}_1 > \rho_1(\rho + \hat{\rho})$$

and coincides with (3.2).

Thus partially stable regime in the original model Σ implies the positive recurrence of corresponding single orbit system: $\mathsf{E}\hat{\mathsf{B}} < \infty$, and we can apply the regenerative method of confidence estimation for mean number of orbit customers in $\hat{\Sigma}$. Define by $\hat{r}$ the mean orbit size, by $\hat{r}_k$ and $\hat{\Delta}_k$ the corresponding estimators obtained with the regenerative method for the system $\hat{\Sigma}$ exactly as in (4.3). (Note k defines the number of regeneration cycles in $\hat{\Sigma}$).

Next our goal is to validate the accuracy of interval $[r_k \pm \Delta_k]$, comparing it with $[\hat{r}_k \pm \hat{\Delta}_k]$ under assumption that conditions (3.2) and (3.3) hold true. Note that (3.3) does not influence to the stability of $\hat{\Sigma}$ and the regenerative estimation is applicable even if (3.3) is violated, but in this case original model Σ does not converge to $\hat{\Sigma}$ the comparison of obtained intervals have no sense. Note that under conditions

$$\begin{aligned} \hat{\rho}_1 &> \rho_1(\rho + \hat{\rho}), \\ \rho &\leq \hat{\rho}_2/(\rho_2 + \hat{\rho}_2) \end{aligned}$$

the model Σ is strictly stable, see [5].

5. Simulations

We assume exponential distributions of service times and fix the following values:

$$\lambda_1 = 4,\ \lambda_2 = 1, \quad \mu_1 = 8,\ \mu_2 = 4,$$

thus

$$\rho_1 = 0.5,\ \rho_2 = 0.25,\ \rho = 0.75.$$

5.1. Partial stability region

We define $\alpha_1 = 20$, $\alpha_2 = 2$, which implies

$$\hat{\rho}_1 = 3.125,\ \hat{\rho}_2 = 0.500, \hat{\rho} = 3.625.$$

Note that initial values of parameters were arbitrary chosen inside the partly stable region to provide the fulfilness of conditions (3.2) and (3.3). Next we consider $n = 100\,000$ arrivals and simulate both systems Σ and $\hat{\Sigma}$. All the experiments were implemented in RStudio development environment. We obtained $k_1 = 6083$ regenerations in the original system, $k_2 = 6020$ regeneration in the single orbit system. Average orbit sizes as follows: $r_{k_1} = 3.66$, $\hat{r}_{k_2} = 3.68$. The comparison of confidence intervals obtained by regeneration method is presented on Figure 1. The results for both systems almost coincide: $\Delta_{k_1} \approx \hat{\Delta}_{k_2} = 0.36$.

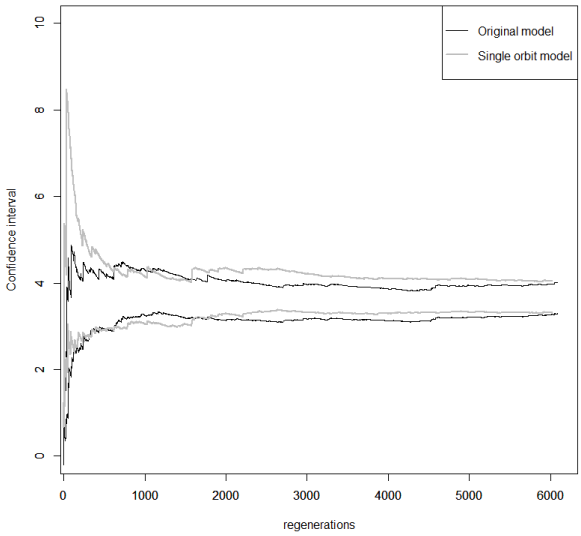


Figure 1. Mean orbit size in Σ and $\hat{\Sigma}$, $\alpha_1 = 20$, $\alpha_2 = 2$.

5.2. Orbit-2 stability border

Next we set the value $\alpha_2 = 2.8$, while $\alpha_1 = 20$. Thus in comparison with the first example we decrease the difference between two parts of the inequality (3.3) and go closer to the border of stability region for the system Σ. Note that in this case the input stream in $\hat{\Sigma}$ is more intensive. We obtain $k_1 = 4184$, $k_2 = 3957$, $r_{k_1} = 4.32$, $\hat{r}_{k_2} = 4.79$. The accuracy difference is more notable: $\Delta_{k_1} = 0.38$, $\hat{\Delta}_{k_2} = 0.45$. Confidence intervals for the considered parameters are presented on Figure 2.

With the growth of the second orbit rate the difference between two models become more significant.

5.3. Instability border

In this example we define $\alpha_1 = 11$, $\alpha_2 = 2$. Thus we touch on the condition (3.2) and move closer to the instability border for the model $\hat{\Sigma}$. (Note that in case the condition (3.2) is violated and (3.3) holds, both orbits in Σ go to infinity, see [5].)

We obtained rare (in comparison with previous cases) regenerations $k_1 = 1162$, $k_2 = 1081$. Note that all simulations are based on $n = 100\,000$ arrivals. Less number of regeneration cycles provide less accurate intervals $r_{k_1} = 18.93$, $\Delta_{k_1} = 3.89$, $\hat{r}_{k_2} = 21.91$, $\hat{\Delta}_{k_2} = 3.91$.

Remind that in all presented examples the conditions (3.2) and (3.3) hold. We started from $\alpha_1 = 20$, $\alpha_2 = 2$ and then explored the cases $\alpha_2 \uparrow$ and $\alpha_1 \downarrow$. Namely in examples B and C we decreased the differences in two parts of inequalities (3.2)

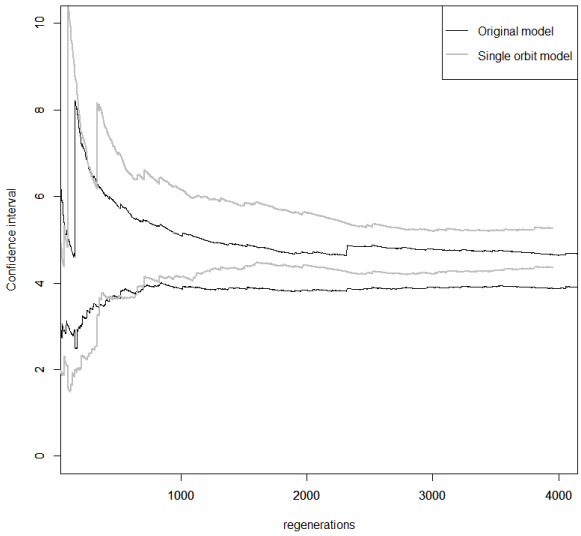


Figure 2. Mean orbit size in Σ and $\hat{\Sigma}$, $\alpha_1 = 20$, $\alpha_2 = 2.8$.

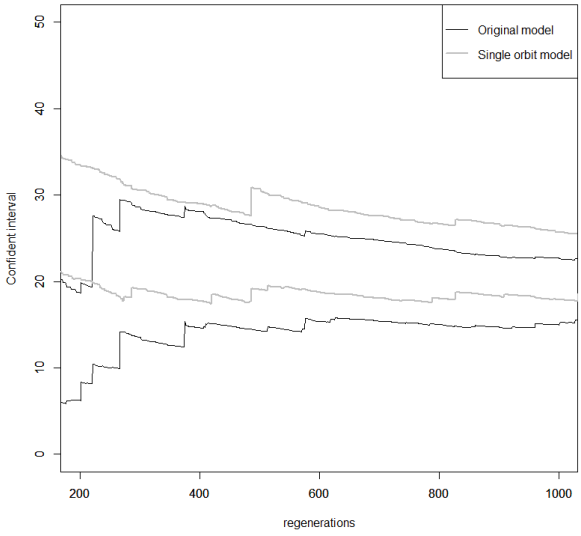


Figure 3. Mean orbit size in Σ and $\hat{\Sigma}$, $\alpha_1 = 11$, $\alpha_2 = 2$.

and (3.3), respectively. Note that in cases $\alpha_2 \downarrow$ and $\alpha_1 \uparrow$ the model Σ converges to $\hat{\Sigma}$ and confidence intervals obtained for mean orbit sizes for both system almost coincide (as on Figure 1).

6. Conclusion

In this paper we study two-class retrial model with constant retrial rates in partially stable regime. In spite of the model under consideration is not stable, we analyze the positive recurrent class-1 orbit size process and apply the regenerative method to construct confidence interval for mean number of class-1 orbit customers. The simulation results correspond with confidence intervals obtained for strictly stable single-orbit model. Thus we illustrate that partially stable case allows to provide accurate confidence estimation.

References

[1] J. Artalejo, A. Gomez-Corral, in: Retrial Queueing Systems: A Computational Approach, Cham: Springer, 2008.

[2] J. Artalejo, T. Phung-Duc: *Single server retrial queues with two way communication*, Applied Mathematical Modelling 37 (2013), pp. 1811–1822.

[3] S. Asmussen, in: Applied probability and Queues. 2nd edn. New York: Springer, 2003.

[4] S. Asmussen, P. Glynn, in: Stochastic Simulation: Algorithms and Analysis, New York: Springer-Verlag, 2007.

[5] K. Avrachenkov, M. E., R. Nekrasova: *Stability analysis of two-class retrial systems with constant retrial rates and general service times*, ArXiv abs/2110.09840 (2021).

[6] K. Avrachenkov, E. Morozov, B. Steyaert: *Sufficient stability conditions for multi-class constant retrial rate systems*, Queueing Systems 82.1-2 (2016), pp. 149–171.

[7] K. Avrachenkov, P. Nain, U. Yechiali: *A retrial system with two input streams and two orbit queues*, Queueing Systems 77.1 (2014), pp. 1–31.

[8] K. Avrachenkov, U. Yechiali: *Retrial networks with finite buffers and their application to internet data traffic*, Probability in the Engineering and Informational Sciences 22.4 (2008), pp. 519–536.

[9] I. Dimitriou: *A queueing system for modeling cooperative wireless networks with coupled relay nodes and synchronized packet arrivals*, Performance Evaluation 114 (2017), pp. 16–31.

[10] G. Falin: *A survey of retrial queues*, Queueing systems 7.2 (1990), pp. 127–167.

[11] G. Fayolle, V. A. Malyshev, M. V. Menshikov, in: Topics in the Constructive Theory of Countable Markov Chains. 1st edn. Cambridge University Press, 1995.

[12] A. Law, D. Kelton, in: Simulation Modeling and Analysis. 5th edn. New York: McGraw-Hill, 2014.

[13] E. Morozov: *A multiserver retrial queue: Regenerative stability analysis*, Queueing systems 56 (2007), pp. 157–168.

[14] E. Morozov, R. Delgado: *Stability analysis of regenerative queues*, Automation and Remote control 70 (2009), pp. 1977–1991.

[15] E. Morozov, T. Phung-Duc: *Regenerative analysis of two-way communication orbit queue with general service time*, Proceedings International Conference Queueing Theory and Network Applications 10932 (2018).

[16] E. Morozov, A. Rumyantsev, S. Dey, T. Deepak: *Performance analysis and stability of multiclass orbit queue with constant retrial rates and balking*, Performance Evaluation 134.10200 (2019).

[17] E. Morozov, B. Steyaert, in: Stability Analysis of Regenerative Queueing Models: Mathematical Methods and Applications, Springer, 2021.

[18] T. Phung-Duc: *Retrial Queueing Models: A Survey on Theory and Applications*, ArXiv abs/1906.09560 (2019), pp. 1–31.

[19] T. Phung-Duc, W. Rogiest, Y. Takahashi, H. Bruneel: *Retrial queues with balanced call blending: analysis of single-server and multiserver model*, Annals of Operations Research 239.2 (2016), pp. 429–449.

Annales Mathematicae et Informaticae
56 (2022) pp. 95–108
DOI: https://doi.org/10.33039/ami.2022.12.003
URL: https://ami.uni-eszterhazy.hu

Generalized Middle-Square Method*

Viktória Padányi, Tamás Herendi

Department of Computer Science, Faculty of Informatics, University of Debrecen
padanyi.viktoria@inf.unideb.hu
herendi.tamas@inf.unideb.hu

Abstract. In this paper, we generalize John von Neumann's Middle-Square Method (MSM) to canonical number systems (CNS). Additionally, we present some observations and statistical tests of the sequences generated by the described generators.

Keywords: pseudorandom number generator, middle square method, canonical number system

AMS Subject Classification: 11K45, 11A63, 11B37, 62-08

1. Introduction

Pseudorandom number generators (PRNG) are often used in solving different theoretical and practical problems. The particular applications expect appropriate properties. The most important properties are the distribution of elements produced by the generators, the low correlation between the consecutive elements, and the large period length. In terms of usage, the speed, the resource requirements, and the qualities of the generators are interesting issues. A general approach for constructing pseudorandom number sequences is the following: the elements of the sequence are computed from the previous elements recursively. Recursion can be resolved by the use of a seed. The next seed is computed iteratively from the preceding seeds, and the random values are extracted from them.

John von Neumann's Middle Square Method is an interesting way to construct uniformly distributed PRNG since this was the first practical random number gen-

*The presented research has been partially supported by the SETIT Project (no. 2018-1.2.1-NKP-2018-00004), which has been implemented with the support provided by the National Research, Development and Innovation Fund of Hungary, financed under the 2018-1.2.1-NKP funding scheme. The research has been partially supported by the construction EFOP-3.6.3-VEKOP-16-2017-00002. The project was supported by the European Union and co-financed by the European Social Fund.

Submitted: October 27, 2022
Accepted: December 3, 2022
Published online: December 28, 2022

erator. In 1946 John von Neumann introduced the method (first published in [11]). It was simple and fast to execute with ENIAC. He used a recursive definition, where the initial value x_0 is some $2k$-digit decimal number. For $n > 0$ he defined $x_n = \lfloor x_{n-1}^2/10^k \rfloor \pmod{10^{2k}}$. The period length of the constructed sequence depends on the initial value. In general, the longest period has length at most 8^{2k}, but very often, it is much shorter. Practically, it is a rather weak generator. If the seed became 0, it is 0 for all consecutive members. N. Metropolis [9] investigated the MSM in binary number systems. He showed that in the case of 20-bit numbers, there are only 13 different cycles. The longest period amongst the 13 cases is 142, which is rather short. It is not obvious to recognize this short period because of the long preperiod.

A more detailed description of PRNGs can be found in [4] and [12].

In this paper, we also deal with canonical number systems. It has long been well-known that positive integers can be represented in a digital way. One of the generalizations of this was studied in [4, p. 189] by D. E. Knuth. He defined a number system for the Gaussian integers similar to the rational integers, where the number system had base $-1+i$. This was later further generalized by I. Kátai and J. Szabó [3], where the existence of other number system bases was proved but still in the Gaussian integers. Later, I. Kátai and B. Kovács in [2], B. Kovács in [6] and B. Kovács and A. Pethő in [7], and [8] even further generalized the definition of the CNS for the ring of algebraic integers. They also proved their existence. In [6], a simple condition is given for the construction of CNSs, but in general, it is not obvious to find them. For the sake of simplicity, we will focus on binary CNSs. A. Kovács [5] presented a complete description of binary canonical number systems of degrees not greater than 8. Later, P. Burcsi and A. Kovács [1] extended the results for CNSs of degrees $9, 10$ and 11. The arithmetic in these number systems is similar to the rational integers with the classical digit representation. However, the calculation of the next digit requires a more difficult reduction operation. Our discussion will focus on binary CNSs.

For instance, let α be a root of the polynomial x^2+x+2, e.g., $-\frac{1}{2}+i\frac{1}{2}\sqrt{7}$. One can prove that in the ring of algebraic integers $\mathbb{Z}[\alpha]$, every number can be written in the canonical form $\gamma = \sum_{i=0}^{h} c_i \alpha^i$, where $c_i \in \{0, 1\}$.

Example 1. Addition in a binary number system

				1	1	0	1	0	1
			+	1	0	1	1	1	1
				2	1	1	2	1	2
			-				1	1	2
				2	1	1	1	0	0
	-	1	1	2					
		-1	-1	0	1	1	1	0	0
+	1	1	2						
	1	0	1	0	1	1	1	0	0

Let $\gamma_1 = 110101$ and $\gamma_2 = 101111$ be 6-bit long binary numbers in the above representation. Then $\gamma = \gamma_1 + \gamma_2$ can be calculated according to Example 1. Here we used the fact that $1\,1\,2$ represents 0.

In the following sections, we define the generalized version of the MSM in binary CNS and analyze some properties of these generators.

2. Definitions and preliminary results

In this chapter, we define some necessary notions and state important results.

Definition 2.1. Let A be a finite set, and u be a sequence over A. We say that $u \in A^\infty$ is periodic with period length $\varrho \in \mathbb{N}$, if there exists $\varrho_0 \in \mathbb{N}$, such that

$$u_{n+\varrho} = u_n \quad \text{for all} \quad n \geq \varrho_0 \ .$$

The smallest ϱ_0 and ϱ with the previous property will be called the preperiod and minimal period length of u, respectively.

If $n \geq \varrho_0$, then the subsequence $u_n, \ldots, u_{n+\varrho-1}$ is called a period of the sequence.

Remark 2.2. Let A be a finite set, $u \in A^\infty$, $0 < k \in \mathbb{N}$ and $F : A^k \to A$. If the sequence satisfies the recurrence defined by $u_n = F(u_{n-1}, \ldots, u_{n-k})$ for all $n \geq k$, then u is periodic with period length $\varrho \leq |A|^k$.

The following definition is the generalization of number systems for complex numbers given by I. Kátai and J. Szabó in [3].

Definition 2.3. Let R be an integral domain, $\alpha \in R$ and $N = \{n_1, \ldots, n_m\} \subseteq \mathbb{Z}$. The pair (α, N) is called a number system in R, if any $\gamma \in R$ has a unique representation in the form $\gamma = \sum_{i=0}^{h} c_i \alpha^i$, where $c_i \in N$ for all $0 \leq i \leq h$ and $c_h \neq 0$, if $h \neq 0$. The number system is called canonical, if $N = \{0, 1, \ldots, m-1\}$.

We will use the notation $L(\gamma, \alpha, N) = h+1$, i.e. the length of the representation of γ in the number system (α, N).

Theorem 2.4. *Let $p \in \mathbb{Z}[x]$ be an irreducible polynomial with $\deg(p) = n$, and $p(x) = a_n x^n + \cdots + a_0$ such that $1 = a_n \leq a_{n-1} \leq \ldots a_0$ and $2 \leq a_0$. Furthermore, let α be a root of p and $N = \{0, 1, \ldots, a_0 - 1\}$. Then (α, N) is a canonical number system in $\mathbb{Z}[\alpha]$.*

Proof. The theorem is proven in a more general setting in [6]. □

Let β be an algebraic number of degree $n \geqq 1$. Then $\beta^{(i)}$ denotes the i^{th} conjugates of β for all $i = 1, \ldots, n$.

Let $\alpha, \gamma \in \mathbb{Q}[\beta]$. For the sake of simplicity, we use the notation

$$|\log|_\alpha \gamma = \max_{1 \leq i \leq n} \frac{\log|\gamma^{(i)}|}{\log|\alpha^{(i)}|} \ .$$

Theorem 2.5. *Let β be an algebraic integer of degree $n \geqq 1$, and let (α, N) be a number system in $\mathbb{Z}[\beta]$. Then there exist effectively computable constants $C_1 = C_1(\alpha, N)$ and $C_2 = C_2(\alpha, N)$ depending only on α and N, such that*

$$|\log|_\alpha \gamma + C_1 \leqq L(\gamma, \alpha, N) \leqq |\log|_\alpha \gamma + C_2 \tag{2.1}$$

holds for every $0 \neq \gamma \in \mathbb{Z}[\beta]$.

Proof. The theorem is proven in [8]. □

Remark 2.6. John von Neumann's MSM uses squaring as the only arithmetic operation. We observe how the length of the numbers changes after squaring.

We fix α and the corresponding CNS, and we use the notation $C_1 = C_1(\alpha, N)$, $C_2 = C_2(\alpha, N)$ and $L(\gamma) = L(\gamma, \alpha, N)$.

For example in the usual binary representation $\alpha = 2$. The length of the binary representation of an integer n can be expressed by

$$L(n, 2) = \lfloor \log_2(n) \rfloor + 1 = \left\lfloor \frac{\log n}{\log 2} \right\rfloor + 1 \ ,$$

which means that $C_1 = 0$ and $C_2 = 1$.

With our simplified notation, equation (2.1) is simplified to

$$|\log|_\alpha \gamma + C_1 \leq L(\gamma) \leq |\log|_\alpha \gamma + C_2 \ . \tag{2.2}$$

Let $\gamma \in \mathbb{Z}[\beta]$ be an algebraic integer with length $L(\gamma)$. By (2.2),

$$|\log|_\alpha \gamma \leq L(\gamma) - C_1 \ , \tag{2.3}$$

and

$$L(\gamma) - C_2 \leq |\log|_\alpha \gamma \ . \tag{2.4}$$

Applying (2.2), (2.3) and (2.4) to the length of γ^2, we obtain

$$\begin{aligned} L(\gamma^2) &\geq |\log|_\alpha \gamma^2 + C_1 = 2|\log|_\alpha \gamma + C_1 \\ &\geq 2(L(\gamma) - C_2) + C_1 = 2L(\gamma) - 2C_2 + C_1 \end{aligned}$$

and

$$\begin{aligned} L(\gamma^2) &\leq |\log|_\alpha \gamma^2 + C_2 = 2|\log|_\alpha \gamma + C_2 \\ &\leq 2(L(\gamma) - C_1) + C_2 = 2L(\gamma) - 2C_1 + C_2 \ . \end{aligned}$$

With the notations $C_3 = C_1 - 2C_2$ and $C_4 = C_2 - 2C_1$, we have

$$2L(\gamma) + C_3 \leq L(\gamma^2) \leq 2L(\gamma) + C_4 \ . \tag{2.5}$$

We should remark that C_1 and C_3 may have negative values. In Section 4, we show some estimates on the values of C_3 and C_4 for different α's.

Since $L(\gamma)$ and $L(\gamma^2)$ are integers, thus C_3 and C_4 can be chosen to be integers without losing precision.

B. Kovács and A. Pethő in [8] prove not only the existence of the constants but also provide a way how to determine them. Their formula is explicit for C_1 but implicit for C_2. Based on the described method, we calculated the values of C_1, C_2, C_3, and C_4 for some polynomials.

By the proof of the Thoerem of [8]

$$C_1 = \min_{1\le i\le n} \frac{\log\left(\left|\alpha^{(i)}\right| - 1\right) - \log(a_0 - 1)}{\log\left|\alpha^{(i)}\right|} \ .$$

For the determination of C_2, one has to compute first some intermediate bounds

$$C_{2,i} = \frac{a_0 - 1}{\left|\alpha^{(i)}\right| - 1} \ .$$

Now, let

$$\Gamma = \left\{ \delta \,|\, \delta \in \mathbb{Z}[\alpha],\ \left|\delta^{(i)}\right| \le C_{2,i} \right\} ,$$

and

$$C_2 = \max_{\delta\in\Gamma} L(\delta, \alpha) \ .$$

In the following, we show the values of the constants for some binary number systems. The structures are defined by $\mathbb{Z}[\alpha]$, where α's are given by their defining polynomials $p(x)$. In these computations, the symbol $\mathbf{i}$ denotes the imaginary unit (everywhere else in the paper, i is an integer).

$p(x) = x^2 + x + 2$:

$$\alpha_{1,\,2} = -\frac{1}{2} \pm \mathbf{i}\frac{1}{2}\sqrt{7} \qquad |\alpha_1| = |\alpha_2| = \sqrt{2}$$
$$C_{2,1} = C_{2,2} \approx 2.41$$
$$C_1 \approx -2.54 \qquad C_2 = 6$$
$$C_3 = -12 \qquad C_4 = 10$$

$p(x) = x^2 + 2x + 2$ (the case of Gaussian integers, considered by Knuth in [4, p. 189]):

$$\alpha_{1,\,2} = -1 \pm \mathbf{i} \qquad |\alpha_1| = |\alpha_2| = \sqrt{2}$$
$$C_{2,1} = C_{2,2} \approx 2.41$$
$$C_1 \approx -2.54 \qquad C_2 = 8$$
$$C_3 = -16 \qquad C_4 = 12$$

$p(x) = x^3 + x^2 + x + 2$:

$$\alpha_1 \approx -1.35 \qquad \alpha_{2,\,3} \approx 0.18 \pm \mathbf{i} \cdot 1.20$$
$$C_{2,1} \approx 2.83 \qquad C_{2,2} = C_{2,3} \approx 4.64$$
$$C_1 \approx -7.85 \qquad C_2 = 13$$

$$C_3 = -31 \qquad C_4 = 27$$

$p(x) = x^4 + x^3 + x^2 + x + 2$:

$$C_{2,1} = C_{2,2} \approx 3.97 \qquad C_{2,3} = C_{2,4} \approx 7.72$$
$$C_1 \approx -16.77 \qquad C_2 = 21$$
$$C_3 = -46 \qquad C_4 = 32$$

In the last two cases, we detailed only some significant steps of the above-mentioned computation.

Related to the previously computed constants, we did some experiments. In Table 2, we collected the results of how the sizes of the squares changed after squaring in the considered canonical number systems. The set of four CNSs is extended by the two rational binary number systems with bases 2 and -2.

We consider all the numbers of 20 to 30 digits. For a given length h, the table contains the distances of the minimal and maximal lengths of the squares from the expected $2h$. Additionally, the average lengths of the squares are presented. The last column displays the theoretical bounds for the corresponding values of distances.

Studying the results, one may conjecture that the minimal and maximal lengths of the squares are considerably closer to the expected value of $2h$ than the analytical computations show. Another suspicion is that the average length of squares is close to $2h$, but increasing the degree of the base α increases the averages.

3. Arithmetic in canonical number systems

Let (α, N) be a CNS and $p(x) = a_n x^n + \cdots + a_0$ be the defining polynomial of α according to Theorem 2.4. The usual arithmetic of integers can be generalized to (α, N). The modified carry computation can be derived from p, described below.

Let $\beta \in \mathbb{Z}[\alpha]$ be the result of some arithmetical operation, and $\beta = \sum_{i=0}^{h} b_i \alpha^i$ is the representation without reduction. If for all $0 \le i \le h$, $b_i \in \{0, \ldots, a_0 - 1\}$ then β is represented in (α, N). Assume now that there exists $0 \le i \le h$ such that $b_i \notin \{0, \ldots, a_0 - 1\}$ and let j be the smallest such integer. Let $c \in \mathbb{Z}$ be such that $b_j = c \cdot a_0 + b'_j$ with $0 \le b'_j < a_0$. Since

$$0 = a_n \alpha^n + a_{n-1} \alpha^{n-1} + \cdots + a_0 \ ,$$

thus

$$\begin{aligned} \beta &= \sum_{i=0}^{h} b_i \alpha^i + c\alpha^j \cdot (a_n \alpha^n + a_{n-1} \alpha^{n-1} + \cdots + a_0) \\ &= \sum_{i=0}^{h'} b'_i \alpha^i \ , \end{aligned}$$

where $b_i = b'_i$ if $0 \leq i < j$, and $b'_j \in \{0, \ldots, a_0 - 1\}$.

In this new representation of β, either all coefficients are in $\{0, \ldots, a_0 - 1\}$ or the smallest k such that $b_k \notin \{0, \ldots, a_0 - 1\}$ satisfies $j < k$. It is proven in [7], that this iteration will terminate in finitely many steps, providing the unique, valid digit expansion of β in (α, N).

Based on the above observation, one can create an algorithm for the arithmetic operations in (α, N), similar to the usual carry computation used for rational integers.

By Theorem 2.4, the results of arithmetic operations have finite representation, whence the carry algorithm will always terminate.

Table 2. Lengths of squares

Length of base numbers

Digits	20	21	22	23	24	25	26	27	28	29	30	T
Defining polynomial: $x - 2$												
Decrease	1	1	1	1	1	1	1	1	1	1	1	1
Increase	0	0	0	0	0	0	0	0	0	0	0	0
Average	39.6	41.6	43.6	45.6	47.6	49.6	51.6	53.6	55.6	57.6	59.6	
Defining polynomial: $x + 2$												
Decrease	3	3	3	3	3	3	3	3	3	3	3	4
Increase	1	1	1	1	1	1	1	1	1	1	1	2
Average	38.9	40.9	42.9	44.9	46.9	48.9	50.9	52.9	54.9	56.9	58.9	
Defining polynomial: $x^2 + x + 2$												
Decrease	8	8	8	8	8	8	8	8	8	8	8	12
Increase	5	5	5	5	5	5	5	5	5	5	5	10
Average	39.6	41.6	43.6	45.6	47.6	49.6	51.6	53.6	55.6	57.6	59.6	
Defining polynomial: $x^2 + 2x + 2$												
Decrease	12	12	12	12	12	12	12	12	12	12	12	16
Increase	9	9	9	9	9	9	9	9	9	9	9	12
Average	40.6	42.6	44.6	46.6	48.6	50.6	52.6	54.6	56.6	58.6	60.6	
Defining polynomial: $x^3 + x^2 + x + 2$												
Decrease	11	14	14	14	14	14	14	14	14	14	14	31
Increase	12	12	12	12	12	12	12	12	12	12	12	27
Average	41.6	43.5	45.5	47.5	49.5	51.5	53.5	55.5	57.5	59.5	61.5	
Defining polynomial: $x^4 + x^3 + x^2 + x + 2$												
Decrease	17	18	18	18	18	18	20	20	20	20	20	46
Increase	21	21	21	21	21	21	21	21	21	21	21	32
Average	43.8	45.6	47.6	49.7	51.9	53.9	55.7	57.7	59.7	61.8	63.8	

4. Generalized Middle-Square Method

Using binary CNSs, we may generalize John von Neumann's MSM.

Let $p(x) \in \mathbb{Z}[x]$ be an irreducible polynomial of degree n, and with coefficients $1 = a_n \leq a_{n-1} \leq \cdots \leq a_0 = 2$. The corresponding CNS has only 2 digits: 0 and 1. For the sake of simplicity, we will call the digits bits and the digit representation of algebraic integers in $\mathbb{Z}[\alpha]$ as a binary representation.

In the design of the generator, we use a seed of $m \in \mathbb{N}$ bits. Similarly, as it is done in the original construction, let u be a sequence over $\mathbb{Z}[\alpha]$ defined by the following:

$u_0 \in$ is a random m-bit number;

if $k > 0$, let

$$u_{k-1}^2 = \sum_{i=0}^{h} b_i \alpha^i \ , \text{ with } b_h \neq 0 \ , \ t = \left\lfloor \frac{h-m}{2} \right\rfloor \text{ and}$$

$$u_k = \sum_{i=0}^{m-1} b_{i+t+1} \alpha^i \ .$$

The value of m should be chosen to be large enough, in particular such that $2m + C_3 > m$, i.e. $m > -C_3$, where C_3 is as defined in section 2.

Another approach is if $t = \lfloor \frac{m}{2} \rfloor$, but then $\frac{m}{2} > -C_3$ should hold.

5. Experimental results

This section provides some experimental results related to the Generalized Middle-Square Method (GMSM). We observe the periodicity properties for several base polynomials, particularly those studied in the previous sections.

Furthermore, some statistical tests – the distributions of moving averages, zero-crossing gaps, and frequency classes – are presented for the GMSM generators, where the arithmetics are derived from the polynomials $x^2 + x + 2$ and $x^4 + x^3 + x^2 + x + 2$. Comparison of the data – both optically and numerically – shows that increasing the degree of the polynomials improves the properties of the generated sequences.

Figure 1 displays the distributions of the moving average of the sequences.

We have initialized the sequences with randomly chosen integers. The sizes of the samples are 10^8. The seeds are 63-bit words, and the pseudorandom values are obtained by a reduction to the 14-bit prefixes (the least significant 49 bits are eliminated). The length of the window for the summation is 100.

We have used the following simple formula to compute the sequence of moving averages:

$$a_k = \frac{1}{100} \sum_{i=k}^{k+100} u_i \ ,$$

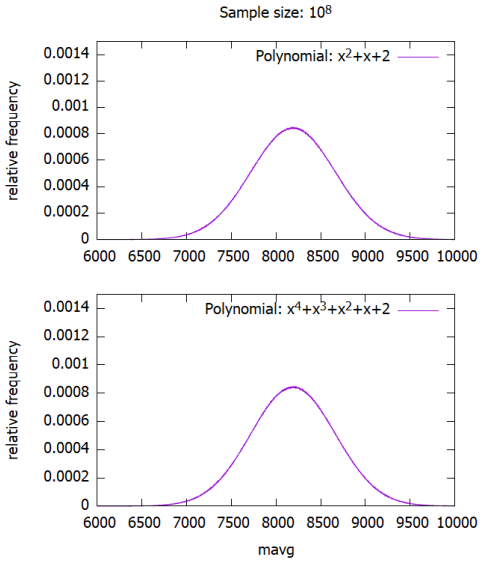


Figure 1. Moving average distribution

where (u_i) is the sequence generated by the GMSM.

Next, we observed the generators' behavior under the random walk test.

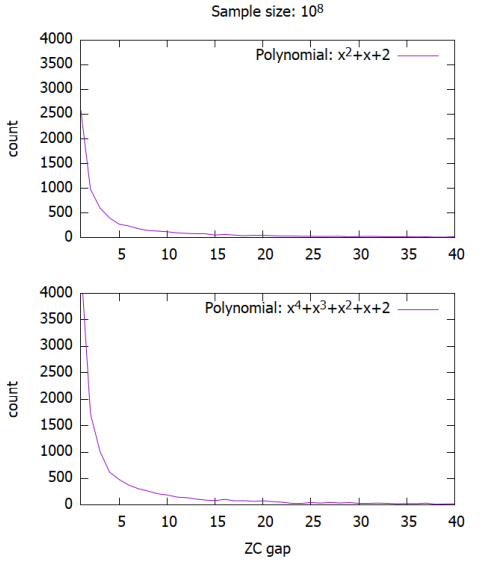


Figure 2. Random walk

The generated sequences are balanced around 0 by a shift with the mean value: $v_k = u_k - E(u)$. Using the new samples, we have computed the cumulative sums:

$$c_k = \sum_{i=0}^{k} v_i \ ,$$

The test calculates the frequency of the lengths of the gaps between consecutive zero crossings of c. The results are presented in Figure 2.

Finally, we have investigated to the distribution of the frequency classes. The values of the sequences are arranged into 2^{14} intervals of equal lengths (again, we reduce the random samples to the 14 most significant bits):

$$\mathrm{U}_i = \left\{ u_k \mid i = \left\lfloor \frac{u_k}{2^{49}} \right\rfloor \right\} \ , \text{ where } i \in \left\{0, \ldots, 2^{14} - 1\right\} \ .$$

Our objective is to describe the probability of the event when the same (reduced) random value appears exactly t times for a given t.

For normalization reasons, the minimum and maximum of the cardinalities are computed:

$$\begin{aligned} \min &= \min\left\{ |\mathrm{U}_i| \mid i = 0..2^{14} - 1 \right\} \text{ and} \\ \max &= \max\left\{ |\mathrm{U}_i| \mid i = 0..2^{14} - 1 \right\} \ . \end{aligned}$$

Figure 3 displays the distributions of the relative frequencies of the cardinalities of U_k.

The horizontal axis is normalized, and the plotted values are calculated according to the following formulas:

$$\begin{aligned} x_t &= \frac{t - \min}{\max - \min} \ , \\ y_t &= \frac{\left|\left\{ i \mid |\mathrm{U}_i| = t, 0 \le i < 2^{14} \right\}\right|}{10^8} \ . \end{aligned}$$

Although the above-presented graphs show good properties of the regarded generators, the investigation of a detailed statistical test provides a more accurate description of the behavior of the sequences. We have tested two of our generators with the NIST Statistical Test Suite (c.f. [10]). The results are summarized in Tables 4 and 5. These two are the MSMs corresponding to the polynomials x^2+x+2 and $x^4 + x^3 + x^2 + x + 2$. We denote them by GMSM1 and GMSM2 in the tables, respectively. In both sequences, we have used a 63-bit seed. The bit sequences for the tests are produced by simply writing the blocks of seeds bit by bit consecutively.

We compared the results with two of the NIST's built-in generators, the LCG and SHA1. The comparison shows that the properties of GMSM sequences are between the two built-in ones.

We used the default parameter adjustments in Table 3.

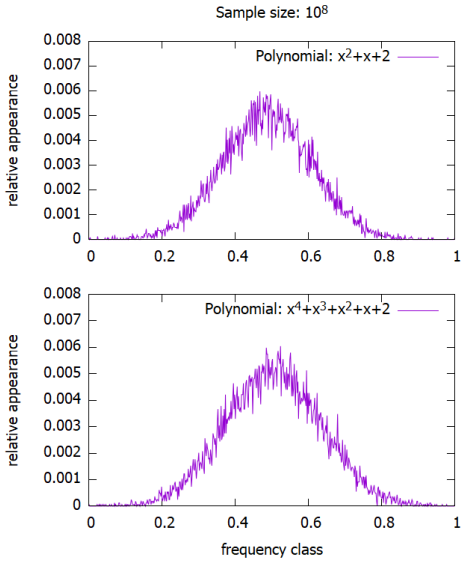


Figure 3. Frequency distribution

Table 3. NIST default settings

Test name	Block length
Block frequency	128
Non-overlapping template	9
Overlapping template	9
Approximate entropy	10
Serial	16
Linear Complexity	500

Both tests have the same arguments: the lengths of the sample sequences are 1000000, and the numbers of independent bitstreams are 1000. The level of acceptance is left to the default 0.01. In Table 4, one can see that both generators have an acceptable uniformity level on average.

Table 5 shows the ratio of the 1000 bitstreams accepted by the tests. Referring to the final report of the NIST test suite, "the minimum pass rate for each statistical test with the exception of the random excursion (variant) test is approximately 0.981819", while "the minimum pass rate for the random excursion (variant) test is approximately 0.979456". Based on this recommendation, we may say that both generators have passed all tests.

Last but not least, in Table 6, we have collected the periodicity properties of the same GMSM sequences as in Table 2.

Again, one block corresponds to the CNS given by the defining polynomial of

its base. The entries are:
- the number of disjoint cycles;
- the maximal length of the cycles;
- the number of the length-1 cycles

for the different seed sizes. The trivial 0-cycle is excluded from the table.

Table 4. NIST test results: p-values

	p-value	
	GMSM1	GMSM2
Frequency	0.574903	0.142872
Block Frequency	0.936823	0.516113
Cumulative Sums	0.225069	0.484351
Runs	0.818343	0.761719
Longest Run	0.015707	0.674543
Rank	0.807412	0.552383
FFT	0.145326	0.368587
Non-Overlapping Template	0.511596	0.501944
Overlapping Template	0.248014	0.825505
Universal	0.152044	0.655854
Approximate Entropy	0.769527	0.353733
Random Excursions	0.292500	0.341976
Random Excursions Variant	0.480915	0.385875
Serial	0.145441	0.236631
Linear Complexity	0.492436	0.347257

Table 5. NIST test results: proportions

	Proportion	
	GMSM1	GMSM2
Frequency	0.9870	0.9890
Block Frequency	0.9890	0.9950
Cumulative Sums	0.9855	0.9890
Runs	0.9880	0.9890
Longest Run	0.9870	0.9900
Rank	0.9870	0.9860
FFT	0.9930	0.9870
Non-Overlapping Template	0.9905	0.9895
Overlapping Template	0.9860	0.9910
Universal	0.9920	0.9920
Approximate Entropy	0.9880	0.9850
Random Excursions	0.9853	0.9930
Random Excursions Variant	0.9866	0.9912

Table 6. All cycles in GMSM sequences

Nontrivial cycles

Digits (seed)	10	11	12	13	14	15	16	17	18	19	20
Defining polynomial: $x-2$											
Cycles	4	4	6	4	9	12	12	10	11	6	12
Max period length	5	5	10	2	56	70	111	203	197	2	142
Stability points	2	3	3	3	3	3	4	4	6	5	6
Defining polynomial: $x+2$											
Cycles	2	6	7	7	11	12	16	11	13	18	18
Max period length	3	3	2	34	10	27	51	30	2	39	4
Stability points	1	3	4	3	5	4	6	5	8	9	8
Defining polynomial: x^2+x+2											
Cycles	3	4	4	2	4	6	3	3	4	9	7
Max period length	2	2	10	19	10	13	34	21	13	256	476
Stability points	2	3	1	1	1	1	1	1	2	2	2
Defining polynomial: x^2+2x+2											
Cycles	2	4	6	5	5	7	5	4	7	12	13
Max period length	1	2	2	5	5	11	20	2	7	24	117
Stability points	2	3	4	2	2	2	2	3	5	9	8
Defining polynomial: x^3+x^2+x+2											
Cycles	10	13	6	6	3	1	5	6	7	11	5
Max period length	5	5	9	5	1	1	7	67	20	165	57
Stability points	8	10	4	5	3	1	3	3	3	3	1
Defining polynomial: $x^4+x^3+x^2+x+2$											
Cycles	5	8	6	5	5	7	10	11	6	6	8
Max period length	13	19	4	12	83	22	57	54	270	125	258
Stability points	2	1	3	3	3	3	6	7	2	2	3

The first block contains test results in the CNS with base 2, i.e., the simple binary representation of non-negative rational integers.

In the second block, the number system is the extension of the previous to the whole set of integers with base -2.

One must remark that even if they have small period lengths, the sequences can be used for pseudorandom number generators because of the long preperiod. Increasing the size of the seed increases the period length and the length of the longest period, but not in a monotonous way.

References

[1] P. Burcsi, A. Kovács: *Exhaustive search methods for CNS polynomials*, Monatshefte für Mathematik 155 (3) (2008), pp. 421–430.

[2] I. Kátai, B. Kovács: *Canonical number systems in algebraic number fields*, Acta Math. Hung. 37.1-3 (1981), pp. 159–164.

[3] I. Kátai, J. Szabó: *Canonical number-systems for complex integers*, Acta Sci. Math. 37 (1975), pp. 255–260, issn: 0001-6969.

[4] D. E. Knuth: *The Art of Computer Programming, Volume 2: Seminumerical Algorithms*, Boston: Addison-Wesley, 1981.

[5] A. Kovács: *Generalized binary number systems*, Annales Univ. Sci. Budapest, Sect. Comp 20 (2001), pp. 195–206.

[6] B. Kovács: *Integral domains with canonical number systems*, Publ. Math. 36.1-4 (1989), pp. 153–156, issn: 0033-3883.

[7] B. Kovács, A. Pethő: *Number systems in integral domains, especially in orders of algebraic number fields*, Acta Sci. Math. 55.3-4 (1991), pp. 287–299, issn: 0001-6969.

[8] B. Kovács, A. Pethő: *On a representation of algebraic integers*, Stud. Sci. Math. Hung. 27.1-2 (1992), pp. 169–172, issn: 0081-6906; 1588-2896/e.

[9] N. Metropolis: *Phase shifts — middle squares — wave equations*, Symposium on Monte Carlo methods, University of Florida (1954), pp. 29–36.

[10] National Institute of Standards and Technology: *A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications: NIST SP 800-22*, 2012, url: https://www.nist.gov/publications/statistical-test-suite-random-and-pseudorandom-number-generators-cryptographic (visited on 02/19/2017).

[11] J. von Neumann: *Various Techniques Used in Connection with Random Digits. In: A.S. Householder, G.E. Forsythe, and H.H. Germond, eds., Monte Carlo Method, National Bureau of Standards*, Appl. Math. 12 (1950), pp. 36–38.

[12] V. Padányi, T. Herendi: *Metaanalysis of Pseudorandom Number Generators*, 23rd annual Spring Wind conference 23 (2020), pp. 474–486.

Annales Mathematicae et Informaticae
56 (2022) pp. 109–121
DOI: https://doi.org/10.33039/ami.2022.12.010
URL: https://ami.uni-eszterhazy.hu

The effectiveness of the Webtable-Datatable Conversion approach*

Katalin Sebestyén[a], **Gábor Csapó**[a], **Mária Csernoch**[b]

[a]University of Debrecen, Doctoral School of Informatics
sebestyen.katalin@inf.unideb.hu
csapo.gabor@inf.unideb.hu

[b]University of Debrecen, Faculty of Informatics
csernoch.maria@inf.unideb.hu

Abstract. Informatics education in Hungary is based on the National Base Curriculum (NAT) and the Frame Curricula. These documents contain the subjects (sciences), the number of classes for each subject and the requirements for each grade. According to the NAT2012, Informatics as a compulsory school subject is introduced in Grade 6. The filemanagement is among the first topics that students must learn according to the Frame Curricula. However, this is not their first encounter with filemanagent, since by the age of 12 most of the students are already active users of digital tools, and associated with the false assumptions of digital natives. Due to the late introduction, the filemanagement is one of the most neglected topics in informatics education. Nevertheless, this is one of the most important topics, since it is essential for further development in handling digital products. Our research group developed the Webtable-Datatable Conversion (WDC) high-mathability method to teach filemanagement. This approach not only focuses on the main file operations but handles real world problems which require firm algorithm construction and datamanagement. The aim of the present study is to measure the effectiveness of the WDC approach with Grade 9 students, where the comparison of groups studying with the traditional and the WDC methods was carried out.

Keywords: K-12 education, filemanagement, knowledge-transfer, informatics education

*This work was supported by the construction EFOP-3.6.3-VEKOP-16-2017-00002. The project was supported by the European Union, co-financed by the European Social Fund.

Submitted: November 23, 2022
Accepted: December 3, 2022
Published online: December 28, 2022

AMS Subject Classification: 03-08, 03-11, 03B47, 97D50

1. Introduction

1.1. The role of filemanagement in studying informatics

According to researchers [5, 6, 26, 28–30] and the Hungarian National Base Curriculum [20], computational thinking is the fourth basic skill alongside the 3Rs (Reading, wRiting and aRithmetic). Consequently – similar the other three skills – computational thinking should be developed from the beginning of organized education [13, 16, 25]. Although, the development of computational thinking skills is emphasized in the National Base Curriculum, in the Frame Curricula [18, 19] – which is created on the bases of the National Base Curricula [20] – is hardly detectable. The focus of the informatics Frame Curricula [18, 19] is reather on teaching the software environments and tools, for the students to be able to navigate in programs. Furthermore, informatics as a compulsory subject is introduced in Grade 6, only one class a week. The primary reasons why the filemanagement does not receive enough attention is due to the extremely low number of classes, the late introduction of the school subject, and the attitude assigned to digital natives [16, 22]. In general, it is assumed that every student know it and use it, and consequently, there is no need to pay attention to this topic.

1.2. High-mathability teaching approaches

The IEEE & ACM report [1] defines three level of mastery, which is in complete accordance with Pólya's [21] concept-based high-mathability problem-solving method [2–4, 8, 9]. The steps are built on each other, so are the levels of mastery: analyses of the problem (1), construction of plan (2), implementation (3), and discussion (4). However, in informatics education, the focus is on the third level of problem solving, ignoring the first and the second – understanding the problem, what we know about the problem and the planning, building algorithm – thus making it impossible to reach the fourth level, the evaluation, the discussion. Overall, the Hungarian Frame Curricula [18, 19] does not pay attention to the development of the students' computer thinking skills and does not support the algorithm building and the schema construction, which play crucial role in cognitive load [27] and ultimatily activating fast and slow thinking effectively [12, 15].

2. WDC method

The Webtable-Datatable Conversion (WDC) [4, 12, 14] method is a high-mathability approach [4, 7, 10], which is based on the use of schemata and building algorithm in the subject of filemanagement. At the beginning of the educational process the teacher raises a problem: how a table available on an webpage (webtable) can be

converted to a datatable for further use in spreadsheet- and/or database- management or programming. WDC is time consuming teaching-learning appoach for developing fundamental skills in informatics. Furthermore, the method heavily relies on the students' knowledge stored in the longterm memory. Considering theses bases, the teacher leads the conversation with coaching techniques [9, 17], where targeted questions are used to help students to set up the characteristics of the data, to understand the problem, and to find their own solutions. In Pólya's terminology the method is entitled guided discovery [21]. Students can also get help in Redmenta [23] where a matching test is built to find operations and the corresponding steps of the algorithms (the tasks are developed by one of our pre-service teachers of informatics). Based on the algorithm, the students complete the steps which primarily are fundamental file operations: save, save as, create, open, close, etc.

The matching tasks (Redmenta) develop students' fast thinking skills [15, 27], based on the schemata build up in long-term memory. It is important that the students do not only follow strict steps, they rather focus on the problem and the problem-solving strategies, otherwise they would not be able to solve the tasks, since they are all different – authentic content. During the conversion process, office applications are used – especially browsers, word processers, and spreadsheet programs. The selection of the program depends on the original sources, the webtables, and the goals of the classes and the projects. Here, we must note that using these programs in the conversion process allow us to lay the fundamental skills to their effective use.

3. Measurement

To quantify and prove the efficiency of the WDC method, our research group tested experiment groups where this novel, high-mathability approach was introduced, and compared their results to control groups where the traditional, low-mathability, tool- and environment-focused methods are used to teach file handling (based on the Frame Curricula). Four hypotheses were formulated to see how students develop using the low-mathability and the newly introduced WDC method.

H1. In the pre-test, there is no significant difference between the results of the experimental and the control groups.

H2. The results of the students in the post-test are significantly higher than in the pre-test.

H3. In the post-test, the experimental group reached significantly higher results.

H4. The rate of development was significantly higher in the experimental group than in the control group.

3.1. Sample

The teaching and the testing process took place in the academic year of 2018/2019, in two high schools in Hungary. All students from Grade 9 formed both the experimental and the control groups. Filemanagement is taught from Grade 6 in the elementary school for every student included in the sample, based on the Frame Curricula [18, 19] and on the local curriculum of the schools. The groups were tested before (pre-test) and after (post-test) the teaching period, however some students were unable to participate in both measurement due various reasons: the teacher of two control groups refused to cooperate, students' illness, and school activities. Consequently, the comparison was based on the results of students who completed both tests. Table 1 shows the number of students who participated in our tests.

Table 1. The sample, the number of students participatin in the tests.

	Experiment group	Control group
Pre-test	30	79
Post-test	35	51
Paired	28	45

3.2. Tasks

The test consisted of six tasks with various number of questions focusing on the knowledge and conscious use of concepts of filemanagement: extensions, file types, editing/saving/opening files – in general, handling files. (Appendix)

Task F1 presents a well-known warning message of Windows operating system, which appears when one wants to change the extension of a data file [11]. The testsheet allowed students to mark more than one answer, but there is only one correct answer. With this liberty of selecting multiple answers, our aim was to measure whether students would realize that there is only one correct answer and whether they can reveal the juxtapositions in the answers. The only correct answer is "Changes what program is associated with the extension, but the file remains usable".

Task F2 was the an open question. Students had to answer the "What happens when we double-click on a document file?" question. This operation is a four-step process, where the expected algorithm is the following:

- checking the extension of the file
- checking the assigned program to the extention
- running the assigned program
- opening the file

In Task F3 students had to provide an answer on how a spreadsheet can be converted into a text file. Students were able to choose the correct answer from the listed options. Task F4 inquired about the cut file operation: "What happens when you cut a file?" In a similar way to the previous tasks, students had to choose their answers from the listed options. We allowed multiple selections even though that there was only one correct answer. In Task F5, students had to decide the types of the listed files, considering their names and extensions. Based on Tasks F1 and F2, we have found that students are not familiar with the definition of extension and their types. The results of the current task support and extend this finding. This can be explained by the widespread use of the File Explorer present in Windows operating systems, where the extensions of the files are hidden by default. We designed Task F6 to have questions about the same knowledge items using differing approaches and phrasing. In this way we could gather data about the conscious choices and reliable knowledge of students. Each question could be answered with the following options: TRUE, FALSE or I DON'T KNOW.

4. Results

4.1. Pre-test

In the pre-test the average results of the experimental (33.73%) and the control (38.02%) groups were almost the same, the statistical analyses showed no significant differences between the groups ($p = 0.0607$) (Figure 1). We examined the results of the tasks separately where also no difference can be detected (Table 2). These outcomes prove H1 hypothesis, between the results of the groups has no differences in the pre-test.

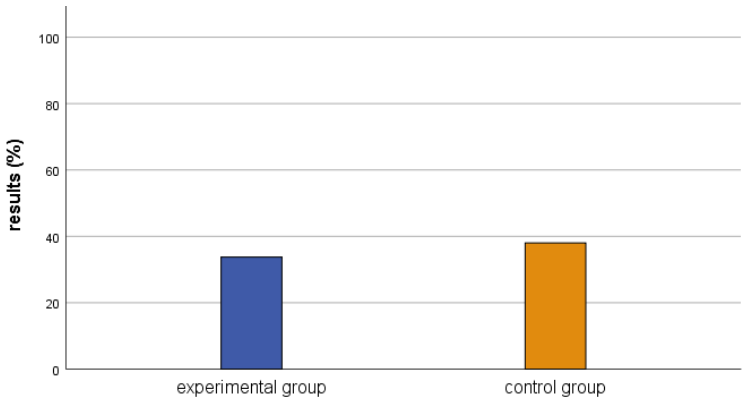


Figure 1. The total results –tasks F1–F6 – of the pre-test by groups (all sample, not only paired).

Table 2. The rate of correct answers for each task in the fileman-agement test including all students.

tasks	experiment group	control group
F1	3.33%	5.06%
F2	19.17%	23.10%
F3	36.67%	34.18%
F4	46.67%	54.43%
F5	36.67%	37.55%
F6	38.33%	44.94%
total	33.73%	38.02%

The students reached extremely low result in some tasks, like F1 and F2 (Table 3). The main reason for this is that incorrect answers were marked along with the correct(s). Lots of the students marked multiple answers, while only one of them was correct. Those answers were accepted as correct where students only marked the correct answer. In the experimental groups 3.33%, in the control group 5.06% of the students marked the correct answer without others. (Table 2).

Table 3. The correct solutions and the proportions of students' answers in Task F2.

	experiment group	control group
extension	0.00%	0.00%
association	0.00%	1.26%
run	0.00%	3.79%
open	75.67%	87.34%
total	19.17%	23.10%

In the other tasks, students completed between 30–50% (Table 2) in average. Despite the higher results, it is clear that the students are not aware of basic definitions and concepts. The computational thinking skills of the students are low, they are not able to explain the process of activities which they carry out frequently.

4.2. Post-test

In the post-test, the number of the students was lower than in the pre-test (Table 1). The experimental group, in almost every task reached significantly higher results than the control group, expect in task F4 (Table 4). Consequently, the total result of the experimental group is significantly higher than of the control group, which proves H2 hypotheses (p=0.0000) (Table 4).

Table 4. The average results (%) of the two groups of students in the post-test.

tasks	experiment group	control group	p
F1	28.57%	7.84%	0.0200
F2	35.00%	24.51%	0.0110
F3	62.867%	21.57%	0.0001
F4	40.00%	47.06%	0.5229
F5	53.33%	36.93%	0.0019
F6	55.95%	45.91%	0.0003
total	50.51%	37.88%	0.0000

In Task F4, similar to the previous tasks, students had to choose their answers from the listed options. We allowed multiple selections even though that there was only one correct answer. In this task, students from the experimental group reached 40%, while the control group score is 47.06%. We must note here that in the experimental group with the exception of one student, everyone marked the correct answer (97.14%). However, those answers cannot be accepted where multiple answers were marked, even though one of them is the correct answer. In contrast, in the control group 84.31% of the students could recognize the correct solution which is less than in experimental group. Table 1 contains the number of students participating in both tests. For the comparison we used and work with the results of those students who participated both tests. The experimental group improved its score except in Task F4 and they reached significantly higher results in the following tasks: F2, F5, F6 (Table 5). The development of the total results is significant compared to the pre-test (p=0.0000). Consequently, the experimental group proves H3 hypothesis.

Table 5. The comparison of the results of the pre- and post-tests.

	experiment group			control group		
tasks	pre-test	post-test	p	pre-test	post-test	p
F1	3.57%	32.14%	0.0087	4.44%	8.89%	0.4204
F2	19.64%	33.93%	0.0028	25.00%	24.44%	0.7100
F3	35.71%	60.71%	0.0698	33.33%	24.44%	0.2901
F4	50.00%	39.29%	0.3262	53.33%	44.44%	0.3770
F5	36.90%	52.38%	0.0050	40.74%	38.15%	0.5557
F6	38.69%	54.76%	0.0009	40.74%	45.18%	0.0965
total	34.14%	49.57%	0.0000	36.98%	37.87%	0.5795

The results of the control group show different pattern, they could not improve their results significantly in any of the tasks (Table 5). In task F3, F4, F5 lower results were obtained compared to the pre-test, nonetheless the differences were not significant. In terms of total results, there is lesser than 1% improvement, which

clearly demonstrate the ineffectiveness of the low-mathability, traditional methods. According to the results of the test, the control group did not prove H2 hypothesis.

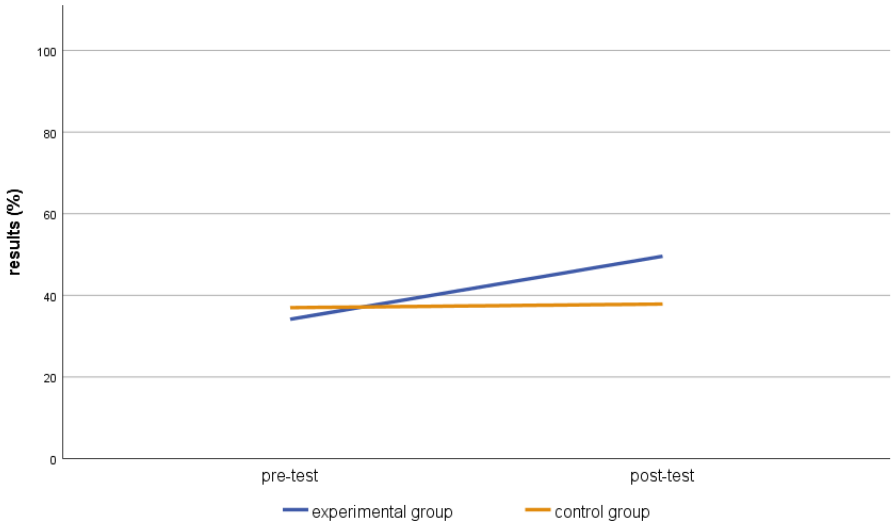


Figure 2. The results in the pre- and post-test.

We measured the difference between the rate of the development, where the experimental group obviously improved to a greater extent (Figure 2). The experimental group started from a lower level (but not significant) and reached a significantly higher level in the post-test. The control group was only able to develop 0.89% during the teaching period, while the experimental group increased its level with 15.43%. Consequently, H4 hypothesis is proved. In general, we can conclude that the high-mathability approach with focusing on schema-construction is more effective than the tool-centered, interface-dependent approaches widely accepted in schools.

5. Misconceptions

The students arrived from several schools, they learned ICT with various methods, this is a reason why they have different IT background knowledge, nonetheless, there were no significant differences between the experimental and the control groups in the pre-test. Therefore, the most common answers of the pre-test were analyzed without grouping. In Task F1 we allowed the multiple selection, however it has only one correct answer. In the pre-test 70.64% of the students marked more options. In the study, we searched for common response pairs, although there were students who marked more than two responses, but the incidence of identity in these groups is low. Based on the pre-test these are the most common response pairs:

- Option 2 with 5
- Option 3 with 6
- Option 3 with 5

Table 6. In the pre-test, the most common students' response pairs in task F1.

pairs	pre-test
2 and 5	32.11%
3 and 6	23.85%
3 and 5	22.93%

The pairs show that the students do not have sufficient knowledge of the concepts of the extension and association. The most common response pairs refer to students' own erroneous experiences and the unquestionable nature of the Windows error message (Table 6) [11]. One explanation to these low resutls that these students learned with interface-based traditional methods which are focused on navigation and the implementation without thinking and problem-solving. Another possibility is that they did not study filemanagement at school, based on the false assumption that as digital natives they already know it. Another common feature of these traditional methods are that students only work with the default extension of the office programs and they use the file explorer, where the file extension is not visible by default.

In the post-test, we analyzed the students' answers by group, where the results of all post-tests in the group were taken into account, not only the paired (Table 1). In the experimental group, the number of the students in the post-test who gave the correct answer increased (Table 4), but the number of multiple responders is still significant (60%). Based on the answers from these students, the following pairs of the answers are the most common (Table 7):

- Option 4 with 6
- Option 1 with 6

Table 7. In the post-test, the most common students' response pairs in task F1 by group.

experiment group		control group	
pairs	results	pairs	results
4 and 6	17.14%	2 and 5	15.68%
1 and 6	14.28%	5 and 6	11.76%

In the experimental group, the most common pair is the 4-6, which contains the correct answer (4), so students have already some knowledge about it, however

not so clear. These students learned with WDC method, consequently, the misunderstanding points out which element of knowledge requires greater focus during the educational process. 70.58% of the students from the control group selected multiple answers. In this group the most common pair is still option 2 and 5, however, a new pair appeared (Table 7). Students in Task F2 did not provide enough answers. The number of students in the post-test who still only know one of the four steps of the process is still high in both group (experimental group 57.14%, control group 86.27%). Therefore, we have not enough data to make a conclusion.

Table 8. The order of preference for the answers to task f3, and its change in the post-test compared to the pre-test.

	experiment group				control group			
	pre-test		post-test		pre-test		post-test	
	pref.	%	pref	%	pref.	%	pref.	%
conversation	4	12.66	4	19.61	4	12.66	4	19.61
export	5	7.59	7	1.96	5	7.59	7	1.96
modifying the extension	2	16.46	2	31.37	2	16.46	2	31.37
google search	6	6.33	–	0	6	6.33	–	0
save as, selecting the new filetype	1	36.71	1	33.33	1	36.71	1	33.33
import	8	2.53	7	1.96	8	2.53	7	1.96
association	7	3.80	5	7.84	7	3.80	5	7.84
save as, changing the filetype manually	3	15.19	3	29.41	3	15.19	3	29.41
online converter	8	2.53	–	0	8	2.53	–	0
open in Notepad	7	3.80	6	3.92	7	3.80	6	3.92

In Task F3, we cannot find pairs to form groups based on the answers. Consequently, we did not look for frequently occurring pairs, but followed the preferences of the students' answers and its changes.

The number of the students from the experimental group who marked the correct answer doubled (Table 8). In contrast, the number of correct answers did not change significantly in the control group. However, the frequency of two responses – modifying the extension; save as, changing the filetype manually – were greatly increased so much so that it equals the number of students who chose the correct option. The knowledge of the control group has become even more fragmented than before. During the educational process in the control group, instead of becoming more accurate, students' knowledge became increasingly burdened with misconceptions, which is a very big problem. In Task F4 many students knew the correct answer, but they chose an extra option. In the pre-test, the most common counterparts to the correct answer is "a copy created of the file" and the "it is moved to the Recycle bin" in both group. This misconception can also be clearly detected in the post-test. The students see the cut operation in two ways:

- by itself: the operation disappears the file during the cut, so students assume that the file is deleted.
- together with another operation: when the concept of the paste operation has merged with cut.

Both methods are needed to eliminate this misconception and to pay attention to it. There is no evidence for misconceptions in Task F5 and F6, only a gap in the students' knowledge is detectable.

6. Conclusion

Filemanagement is one of the most essential topics in informatics and computer sciences. Reliable knowledge cannot be built on uncertain bases, consequently, on this topic greater emphasis should be placed and not be ignored, as has been happening so far.

We have introduced a high-mathability, schema-centered approach entitled WDC. The essence of the method is that tables originated on webpages are converted to datatables primarily through a file conversion processes. The other feature of the method is that real contexts are presented in classes, which increases the motivation of the students.

During the measurement of the effectiveness of the method WDC, we found that in the pre-test, the students, after 3 years of studying informatics in schools, do not have reliable knowledge in filemanagement. Their computational thinking skill is low, they cannot consciously use the tools of the Windows operating system, for example they do not know what happens during cutting operation, what the extension is and what it is for [24].

During the teaching period, the control groups studied with traditional, low-mathability methods, using decontextualized materials, if any, which is the widely accepted approach in educational environments. Our measurement proves that there is no difference between the students' results in the pre- and the post-test, which indicates that the teaching intervention has no effect on the development on the students' skills and knowledge. On the contrary, the experimental group studied with the WDC approach, their result increased in the post-test compared to the pre-test, and the development was found significant.

Based on the results of our measurement, we can conclude that education should not focus on the use of tools, interfaces, and the software environments, but rather on real problem-solving, where tools play a secondary role in the problem-solving process. We have found proof that with the WDC high-mathability approach students can build their knowledge level by level, and they could be solving unknown problems and situations based on their developed concepts and schemata. The analysis of students' responses has drawn attention to a number of misconceptions that provide a good basis for developing teaching-learning methods. It would be worthwhile to explore the cause of misunderstanding, which would make teaching filemanagement more effective.

Our measurement clearly shows the there is a great need for new, effective problem-solving-based approaches in teaching informatics, computer sciences. The requirement of the Frame Curricula cannot be completed with the low-mathability methods widely supported by education systems. The WDC method is an effective alternative for teaching filemanagement, and also lays the fundamentals of the

topics the text- and spreadsheets-management by using authentic sources, real contents. The method based on the concept-based problem-solving approach of Pólya, using the method of guided discovery with an algorithmic focus [4, 24] is proved effective in developing the students' computational thinking skills.

References

[1] ACM/IEEE-CS Joint Task Force on Computing Curricula: *Computer Science Curricula 2013*, tech. rep., ACM Press and IEEE Computer Society Press, 2013, DOI: 10.1145/2534860, URL: http://dx.doi.org/10.1145/2534860.

[2] P. Baranyi, Á. Csapó: *Definition and Synergies of Cognitive Infocommunication*, Acta Polytechnica Hungarica 9.1 (2018), pp. 67–83.

[3] P. Baranyi, Á. Csapó, G. Sallai, Cham, Deutschland: Springer, 2015, DOI: http://dx.doi.org/10.1007/978-3-319-19608-4.

[4] P. Baranyi, A. Gilányi: *Mathability: Emulating and enhancing human mathematical capabilities*, in: 2013 IEEE 4th International Conference on Cognitive Infocommunications (CogInfoCom), 2013, pp. 555–558, DOI: http://dx.doi.org/10.1109/CogInfoCom.2013.6719309.

[5] M. Ben-Ari: *"Bricolage Forever"*, in: Proceedings of the 11th Annual Workshop of the Psychology of Programming Interest Gro-up, Leeds, United Kingdom, 1999.

[6] M. Ben-Ari: *SNon-Myths About Programming*, Communications of the ACM 54.7 (2011), pp. 35–37.

[7] P. Biro, M. Csernoch: *The mathability of computer problem solving approaches*, in: 2016 7th IEEE International Conference on Cognitive Infocommunications (CogInfoCom), IEEE, 2016, pp. 111–114, DOI: https://doi.org/10.1109/CogInfoCom.2016.7804556.

[8] K. Chmielewska, A. Gilányi: *Computer assisted activating methods in education*, in: 10th IEEE International Conference on Cognitive Infocommunications (CogInfoCom), 2019, pp. 241–246, DOI: 10.1109/CogInfoCom47531.2019.9089900.

[9] K. Chmielewska, A. Gilányi: *Educational context of mathability*, Acta Polytechnica Hungarica 15.5 (2018), pp. 223–237.

[10] K. Chmielewska, A. Gilányi, A. Lukasiewicz: *Mathability and Mathematical Cognition*, in: 2016 7th IEEE International Conference on Cognitive Infocommunications (CogInfoCom), IEEE, 2016, pp. 245–250, DOI: https://doi.org/10.1109/CogInfoCom.2016.7804556.

[11] M. Csernoch: *The Stepchild of Informatics Education: File Management*, Academia Letters (2021), pp. 1–4, DOI: http://dx.doi.org/10.20935/AL2295.

[12] M. Csernoch: *Thinking Fast in Computer Problem Solving*, Journal of Software Engineering and Applications 10.1 (2011), pp. 11–40, DOI: https://doi.org/10.4236/jsea.2017.101002.

[13] M. Csernoch, P. Biro, J. Máth, K. Abari: *Testing Algorithmic Skills in Traditional and Non-Traditional Programming Environments*, Informatics in Education 14.2 (2015), pp. 175–197, DOI: https://doi.org/10.15388/infedu.2015.11.

[14] M. Csernoch, E. Dani: *Data-structure validator: an application of the HY-DE mode*, in: 2017 8th IEEE International Conference on Cognitive Infocommunications, IEEE, 2017, pp. 197–202, DOI: https://doi.org/10.1109/CogInfoCom.2017.8268242.

[15] D. Kahneman: *Thinking, Fast and Slow*, New York, United States: Farrar, Straus and Giroux, 2011.

[16] P. A. Kirschner, P. D. Bruyckere: *The myths of the digital native and the multitasker*, Teaching and Teacher Education 67 (2017), pp. 135–142, DOI: https://doi.org/10.1016/j.tate.2017.06.001.

[17] P. A. Kirschner, J. Sweller, R. E. Clark: *Why Minimal Guidance During Instruction Does Not Work: An Analysis of the Failure of Constructivist Discovery, Problem-Based, Experiential, and Inquiry-Based Teaching*, Educational Psychologist 41.2 (2006), pp. 75–86, doi: https://doi.org/10.1207/s15326985ep4102_1.

[18] OFI: *Frame Curricula 2008 In Hungarian: Kerettanterv. 2/2008.(II.8.) számú OKM rendelet – a kerettantervek kiadásának és jóváhagyásának rendjéről*, Magyar Közlöny 20.2 (2008), pp. 1–919.

[19] OFI: *Frame Curricula 2012 In Hungarian: Kerettanterv. 51/2012. (XII. 21.) számú EMMI rendelet – a kerettantervek kiadásának és jóváhagyásának rendjéről*, Magyar Közlöny 177 (2012), pp. 209870–36480.

[20] OFI: *National Base Curriculum in Hungarian: 110/2012. (VI. 4.) Korm. rendelete a Nemzeti alaptanterv kiadásáról, bevezetéséről és alkalmazásáról*, Magyar Közlöny 66 (2012), pp. 10635–10847.

[21] G. Pólya: *How To Solve It: A New Aspect of Mathematical Method*, Princeton, New Jersey: Princeton University Press, 1957, doi: https://doi.org/10.1515/9781400828678.

[22] M. Prensky: *Digital Natives, Digital Immigrants*, MCB University Press 9.5 (2001), doi: https://doi.org/10.1145/1073204.1073229.

[23] *Redmenta*, https://redmenta.com/?media, Accessed: 2021.10.19.

[24] K. Sebestyén: *Students' knowledge in file-management after elementary schoo*, in: Proceedings of the 11th International Conference on Applied Informatics, Eger, Hungary, 2020, pp. 296–308.

[25] A. Settle, B. Franke, R. Hansen, F. Spaltro, C. Jurisson, C. Rennert-May, B. Wildeman: *Infusing computational thinking into the middle- and high-school curriculum*, in: ITiCSE '12: Proceedings of the 17th ACM annual conference on Innovation and technology in computer science education, Haifa, Israel: ITiCSE, 2012, pp. 22–27, doi: https://doi.org/10.1145/2325296.2325306.

[26] E. Soloway: *Should we teach students to program?*, Communications of the ACM 36.10 (1993), pp. 21–25.

[27] J. Sweller, S. K. Paul Ayres: *Cognitive Load Theory*, New York, United States: Springer, 2011, doi: http://dx.doi.org/10.1007/978-1-4419-8126-4.

[28] M. M. Syslo, A. B. Kwiatkowska: *Informatics for All High School Students: A Computational Thinking Approach*, Informatics in Schools. Sustainable Informatics Education for Pupils of all Ages 7780.3 (2013), pp. 43–56, doi: https://doi.org/10.1007/978-3-642-36617-8_4.

[29] J. M. Wing: *Computational thinking*, Communications of the ACM 49.3 (2006), pp. 33–35.

[30] J. M. Wing: *Computational thinking and thinking about computing*, Philosophical Transactions of the Royal Society a Mathematical, Physical and Engineering Sciences 366.1881 (2008), pp. 3717–3725, doi: https://doi.org/10.1098/rsta.2008.0118.

Annales Mathematicae et Informaticae
56 (2022) pp. 122–133
DOI: https://doi.org/10.33039/ami.2022.12.002
URL: https://ami.uni-eszterhazy.hu

Sensitivity analysis of a single server finite-source retrial queueing system with two-way communication and catastrophic breakdown using simulation

János Sztrik, Ádám Tóth

Faculty of Informatics, University of Debrecen, Debrecen, Hungary
sztrik.janos@inf.unideb.hu
toth.adam@inf.unideb.hu

Abstract. In this paper, a finite-source retrial queueing system with two-way communication is investigated with the help of a simulation program of own. If a randomly arriving request from the finite-source finds the single server idle its service starts immediately, otherwise it joins an orbit from where it generates retrial/repeated calls after a random time. To increase the utilization of the server when it becomes idle after a random time an outgoing request is called for service from an infinity source. Upon its arrival if the server is busy, it goes to a buffer and when the server becomes idle again its service starts immediately. requests arriving from the finite-source and orbit are referred to as primary or incoming ones while requests called from the infinite source are referred to as secondary or outgoing requests, respectively. The service times of the primary and secondary requests are supposed to be random variables having different distributions. However, randomly catastrophic failures may happen to all the requests in the system, that is from the orbit, the service unit, and the buffer. In this case, the primary requests return to the finite-source, and the secondary ones are lost. The operation of the system is restored after a random time. Until the restoration is finished no arrivals and service take place in the system. All the above-mentioned times are supposed to be independent random variables.

The novelty of this paper is to perform a sensitivity analysis of the failure and restoration/repair times on the main characteristics to illustrate the effect of different distributions having the same average and variance value. Our aim is to determine the distribution of the number of requests in the system, the average response time of an arbitrary primary request without successful

Submitted: October 21, 2022
Accepted: December 3, 2022
Published online: December 28, 2022

service, also the average response time of an arbitrary and successfully served primary request, the total utilization of the service unit, or the probability that a primary request leaves the system without successful service because of a catastrophic event. Results are illustrated graphically obtained by our simulation program.

Keywords: finite-source queueing, two-way communication, catastrophic failure, restoration, sensitivity analysis, characteristics, simulation

1. Introduction

Retrial queues with two-way communication arose as stochastic models of call centers, where the operator can provide both inbound/incoming and outbound/outgoing calls. The idea of call blending is to improve the productivity of call centers by reducing the idle time of an operator was investigated among others in [2, 3, 5], and references cited in them.

However, from a practical point of view, it is also important to investigate situations where the server is not always able to serve the requests. There are many models and assumptions about the distribution of the operation and restoration time of the server. In case of a breakdown, there are many options corresponding to the behavior of request under service and the request generation process. In this paper, we deal with catastrophes, sometimes called disasters or negative requests which clear all the requests from the service facility, orbit, buffer, and stop the arrivals of the requests. The interested reader is referred to among others [1, 7, 8] and references cited in them.

In our earlier papers we dealt with finite-source single server two-way communication systems with an unreliable server under different repair options and request generation processes. With the help of simulation, the main characteristics were obtained and sensitivity analysis was carried out corresponding to failure and repair time distributions, see [9–11].

The primary aim of the present paper is to carry out a sensitivity analysis of the time of catastrophe and restoration/repair on the main characteristicsto illustrate the effect of different distributions having the same average and variance value. Our goal is to determine the distribution of the number of requests in the system, the average response time of an arbitrary primary request without successful service, also the average response time of arbitrary and successfully served primary request, the total utilization of the service unit, or the probability that a primary request leaves the system without successful service because of a catastrophic event. Results are illustrated graphically obtained by our simulation program.

2. System model

Figure 1 shows the behavior of the system with the aim that we are interested in investigating the effect of the catastrophes on the main characteristics. That

is the reason that we assume exponentially distributed random variables except the distribution of the time of disaster. N sources generate requests after an exponentially distributed time with parameter λ independently of each other. If an arriving request finds the single server idle its service starts immediately, the services time is supposed to be exponentially distributed with parameter μ_1. If the serves is busy the call joins an orbit from where it generates retrial/repeated calls after an exponentially distributed time with parameter ν. To increase the utilization of the server when it becomes idle after an exponentially distributed time with parameter λ_2 an outgoing request is called for service from an infinity source. Upon its arrival, if the server is busy, it goes to a buffer and when the server becomes idle again its service starts immediately. The service time of this type of request is supposed to be exponentially distributed with parameter μ_2. Requests arriving from the finite-source and orbit are referred to as primary or incoming ones while requests called from the infinite source are referred to as secondary or outgoing requests, respectively.

However, randomly catastrophic failures may happen clearing all the requests in the system, that is from the orbit, the service unit, and the buffer. In this case, the primary requests return to the finite-source, and the secondary ones are lost. The operation of the system is restored after an exponentially distributed time with parameter γ_2. Until the restoration is finished no arrivals and service take place in the system. All the above-mentioned times are assumed to be independent random variables. Catastrophes can take place according to gamma, hypo-exponential, hyper-exponential, Pareto and lognormal distribution selecting their parameters to have the same average value.

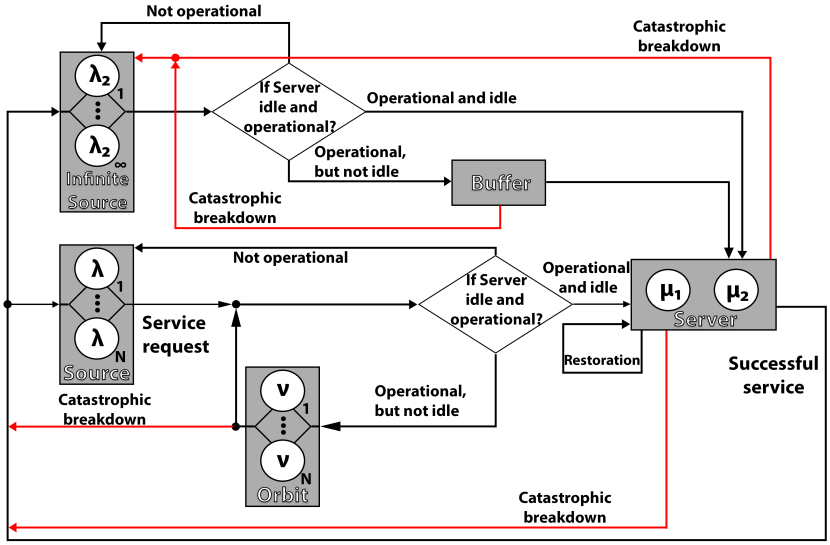


Figure 1. System model.

3. Simulation results and examples

We applied the simulation approach to obtain all the desired characteristics. Due to the simulation, we can deal with generally distributed random variables representing different times that occurred in the model construction. Expect the case when all the random variables are exponentially distributed it is very difficult, if not impossible to get an analytical solution to the characteristics. The estimation is carried out by applying a statistical package in which the method of batch averages is used, see [6]. First, we deal with exponentially distributed failure time with parameter γ_1 and show the effect of the failure rate on the probability that a primary request leaves the system without successful service, see Table 2. Then we turn our attention to generally distributed failure times when the (CV) squared coefficient of variation which is defined as variance/(square of average) is greater or less than one. In both cases we consider distributions with the same average and variances to show the effect of the particular distribution on some of the characteristics.

We must admit by choosing different input parameters our aim is to show how the system behaves and they are not realistic values since we do not have data for this type of system. In this phase the paper is more theoretic than practical.

3.1. Exponentially distributed failure times

In this part, the failure time is assumed to be exponentially distributed with parameter γ_1. The other input parameters are given in Table 1. This model was treated by the help of a software package called MOSEL (MOdeling, Specification and Evaluation Language) and served as a validation for the simulation, see [4].

Table 1. Numerical values of model parameters for exponentially distributed failure time.

N	λ	λ_2	μ_1	μ_2	ν	γ_2
100	0.02	0.5	1	2.5	0.01	1

Table 2. Probability that a primary request departs because of a catastrophic event.

γ_1	P(departure)
0.00001	0.002113
0.01	0.535419
0.1	0.724697

It should be mentioned that even for a small failure intensity the probability of departure is not negligible. In addition, in Figure 2 we can see how the distribution of the number of primary requests changes as the failure rate increases. In the case

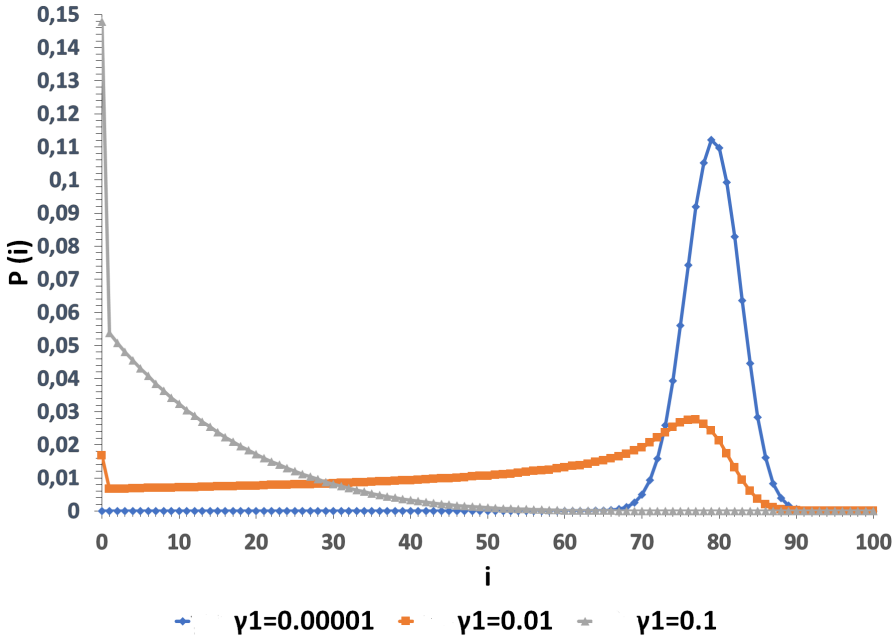


Figure 2. Distribution of the number of primary requests in the system.

of a very small value the distribution graph is similar to a normal distribution, but as we increase the failure rate the distribution is unknown.

3.2. Different distributions of failure time of the system, CV is greater than one

This part is devoted to the sensitivity analysis of the characteristics corresponding to the distribution of failure times. Table 3 shows the used parameter setting and Table 4 collects the values of parameters in the case of gamma, hyper-exponential, lognormal, and Pareto distributions. We assume that $CV > 1$ and to perform a valid comparison both the average value and variance are the same using different parameters' values.

Table 3. Numerical values of model parameters.

$\mathbf{N}$	$\boldsymbol{\lambda_2}$	$\boldsymbol{\mu_1}$	$\boldsymbol{\mu_2}$	$\boldsymbol{\nu}$	$\boldsymbol{\gamma_2}$
100	0.5	1	2.5	0.01	1

The steady-state probability of the number of primary requests in the systems is presented in Figure 3 when $\lambda = 0.02$. Having the same average and variance,

Table 4. Parameters of failure time.

Distribution	Gamma	Hyper-exponential	Pareto	Lognormal
Parameters	$\alpha = 0.31225$ $\beta = 0.05588$	$p = 0.3619707$ $\lambda_1 = 0.1295528$ $\lambda_2 = 0.2283569$	$\alpha = 2.145538$ $k = 2.9835251$	$m = 1.0027833$ $\sigma = 1.1981970$
average	5.588			
Variance	100			
Squared CV	3.2024857438			

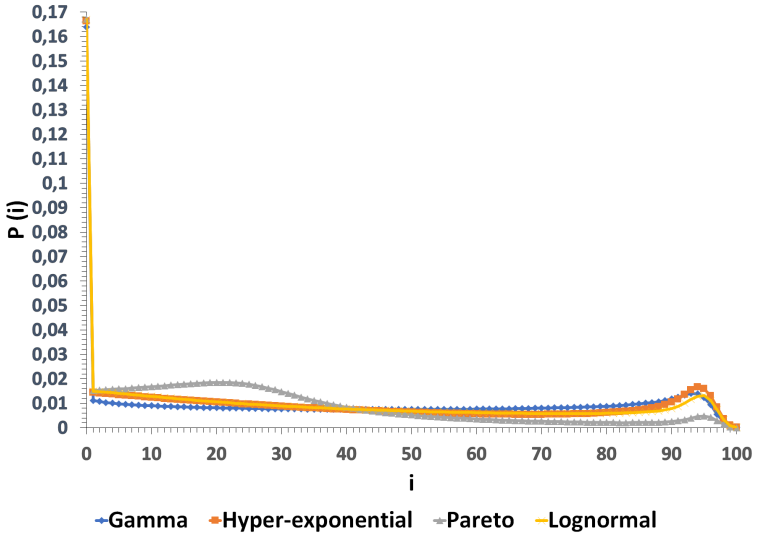


Figure 3. Distribution of the number of primary requests in the system.

the obtained results vary from each other which is especially true in the case of the Pareto distribution. This figure illustrates the impact of the selected distribution on the operation of the system, as was expected.

In Figures 4, 5 the average response time of a primary request and a primary request without successful service can be seen as the function of the arrival rate λ. Essential differences can be observed which is due to the distributions. Naturally, the average response time of requests without successful service should be greater as they leave the system because of catastrophes. Some of them can be in the orbit and one under service. Since the average failure time is 5.588 we expected that all the average response times are less than this value. However, it is true only for the Pareto distribution. It also looks surprising that three averages first increasing then decreasing, while in the Pareto case it is increasing. During several simulation runs, we realized that the behavior of the systems heavily depends on the variance of the failure time and the other input parameters of the system. Our explanation for the unexpected higher average response time is the following.

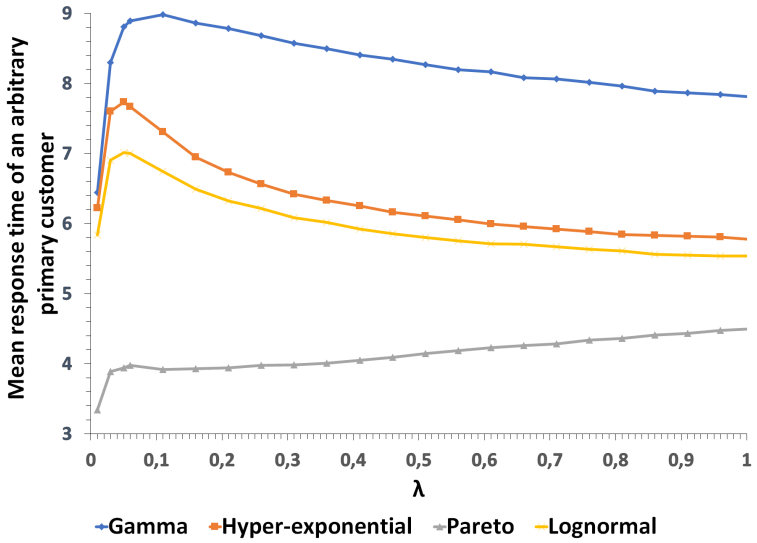


Figure 4. Average response time of a primary request.

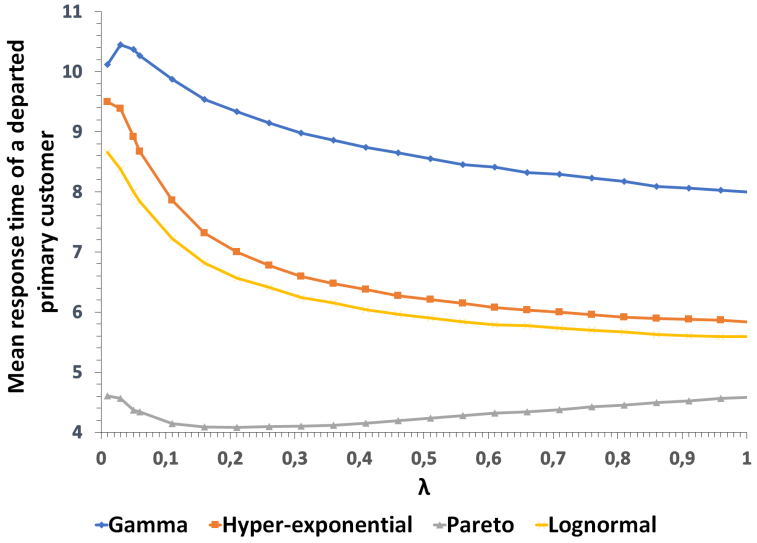


Figure 5. Average response time of a primary request without service.

Since the standard deviation of the operation time is almost two times higher than its average there will be short operation times in which there are no requests in the system, and there are long operation times with high response times. Thus the average response time can be greater than the average operation time. The

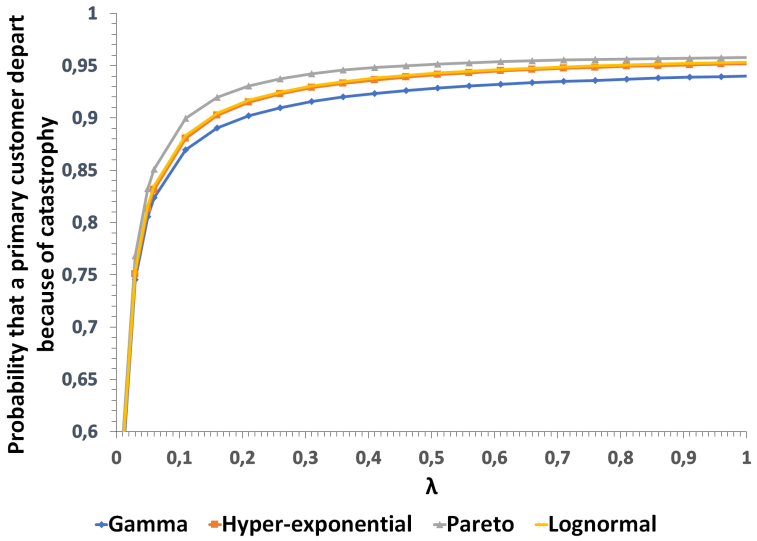


Figure 6. Probability that a primary request departs.

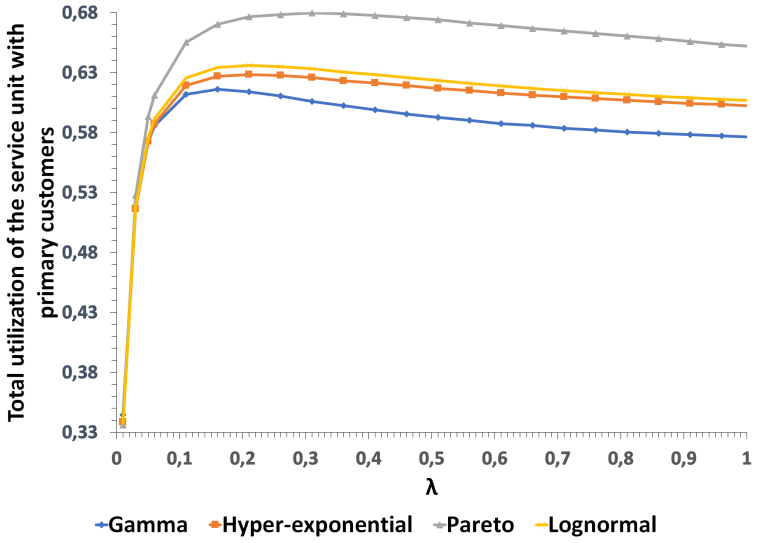


Figure 7. Total utilization w.r. primary requests.

maximum of the average happens only at special parameter setup.

Figure 6 shows the probability that a request departs from the system due to the catastrophe. There are differences between the distributions and of course the probability is an increasing function of the arrival rate from the source since more and more requests are in the system when a catastrophe happens.

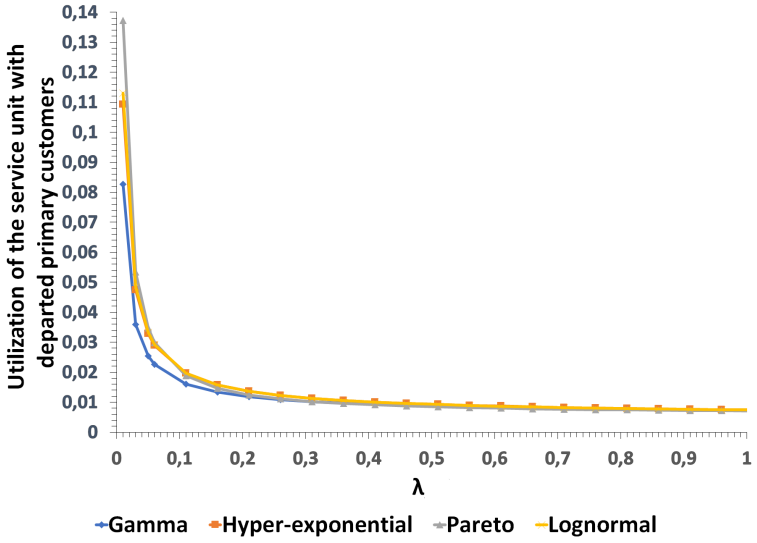


Figure 8. Total utilization w.r. primary requests without service.

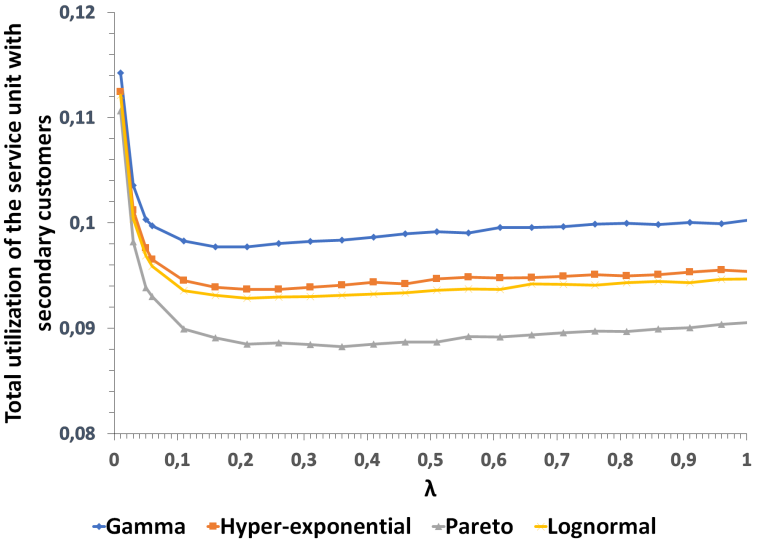


Figure 9. Total utilization w.r. secondary requests.

In Figures 7, 8, 9 utilization of the server corresponding to different types of requests is illustrated. As usual, the utilization of the server with respect to a certain type of request is defined as the probability that the server is busy with that type of request, respectively. There is a very special property of finite-source

retrial queues, namely under special parameter setup the mean response time of a customer has maximum as the function of arrival rate from the source. We could find such a parameter setup that there is maximum of the utilization for the primary requests which includes requests with successful and without successful service, see Figure 7. In the first phase due to the increasing number of requests the utilization increases, but after a certain point due to the catastrophes many requests depart from the system and the utilization decreases.

Figure 8 shows the utilization corresponding to the departed requests due to the catastrophes. Since the number of requests in the system increases as the function of the arrival rate λ more and more requests depart the systems because of the failure, hence the utilization decreases. As we can observe this measure is almost the same for all distributions.

Finally, Figure 9 shows the utilization of the server with respect to the secondary requests invited when the server is idle. The behavior can be explained by the catastrophes since the server in this case is idle and there is more chance for a secondary request to occupy the server.

3.3. Different distributions of failure time of the system, CV is less than one

This part is devoted to the sensitivity analysis of the characteristics with respect to the distribution of failure times. Table 3 shows the used parameter setting and Table 5 collects the values of parameters in the case of gamma, hypo-exponential, lognormal, and Pareto distribution. $CV < 1$ and both the average value and variance are the same using different parameters' values.

Table 5. Parameters of failure time.

Distribution	**Gamma**	**Hypo-exponential**	**Pareto**	**Lognormal**
Parameters	$\alpha = 1.2320819$	$\mu_1 = 0.2$	$\alpha = 2.4940153$	$m = 1.423548$
	$\beta = 0.2204778$	$\mu_2 = 1.7$	$k = 3.3475773$	$\sigma = 0.7708627$
average	5.588			
Variance	25.3460207612			
Squared CV	0.811634349			

Due to the lack of pages, we cannot show the same characteristics as we presented before. We can summarize the findings as follows. The average response times are not greater than the average operation time due to the smaller variance of the operation time. All the other characteristics show similar behavior with fewer differences between the different distributions. In general, performing several simulation runs we observed that the variance of the response times of requests behave similar way as the variance of the operation time either $CV > 1$ or $CV < 1$. One of the advantages of the simulation approach is that we can estimate any of the characteristics giving not only expected values but variances, too.

4. Conclusion

A finite-source retrial queueing system with two-way communication was investigated with the help of simulation. We were interested in carrying out a sensitivity analysis of the failure and restoration/repair times on the main characteristics to illustrate the effect of different distributions having the same average and variance value. We aimed to determine the distribution of the number of requests in the system, the average response time of an arbitrary primary request without successful service, also the average response time of arbitrary and successfully served primary request, the total utilization of the service unit, or the probability that a primary request leaves the system without successful service because of a catastrophic event. Results were illustrated graphically and some explanations were given. The scientific message of the this paper is following: from earlier papers published in different high level journals it can be seen that systems with catastrophic failures are important and needs investigations. The authors are not aware of any papers with two-way communications with this type of failures. In our opinion allowing non-exponentially distributed operation times the analytic solution is hopeless. The only way is the simulation method. It is a natural question to ask how the characteristics of the system depends on the distribution of the operation time assuming the same first two moments, respectively. That was our strong motivation and we are confident that this paper is a valuable contribution to this topic.

Acknowledgment The authors are grateful to the reviewers for their valuable comments and suggestions which improved the quality of the paper.

References

[1] S. R. Chakravarthy: *A catastrophic queueing model with delayed action*, Applied Mathematical Modelling 46 (2017), pp. 631–649.

[2] V. Dragieva, T. Phung-Duc: *Two-Way Communication $M/M/1//N$ Retrial Queue*, in: International Conference on Analytical and Stochastic Modeling Techniques and Applications, Springer, 2017, pp. 81–94.

[3] J. Kim, B. Kim: *A survey of retrial queueing systems*, Annals of Operations Research 247.1 (2016), pp. 3–36, issn: 1572-9338.

[4] A. Kuki, T. Bérczes, J. Sztrik: *Modeling two-way communication systems with catastrophic breakdowns*, in: Information Technologies and Mathematical Modelling. Queueing Theory and Applications, in Press, Springer, 2022.

[5] A. Kuki, J. Sztrik, Á. Tóth, T. Bérczes: *A Contribution to Modeling Two-Way Communication with Retrial Queueing Systems*, in: Information Technologies and Mathematical Modelling. Queueing Theory and Applications, Springer, 2018, pp. 236–247.

[6] A. M. Law, W. D. Kelton: *Simulation modeling and analysis*, McGraw-Hill New York, 1991.

[7] K. Li, J. Wang: *Equilibrium balking strategies in the single-server retrial queue with constant retrial rate and catastrophes*, Quality Technology & Quantitative Management 18.2 (2021), pp. 156–178.

[8] S. Subramanian et al.: *A Stochastic Model for Automated Teller Machines Subject to Catastrophic Failures and Repairs*, Queueing Models and Service Management 1.1 (2018), pp. 75–94.

[9] J. Sztrik, Á. Tóth, Á. Pintér, Z. Bács: *Simulation of finite-source retrial queues with two-way communications to the orbit*, in: International Conference on Information Technologies and Mathematical Modelling, Springer, 2019, pp. 270–284.

[10] J. Sztrik, Á. Tóth, Á. Pintér, Z. Bács: *The Simulation of Finite-Source Retrial Queueing Systems with Two-Way Communications to the Orbit and Blocking*, in: International Conference on Distributed Computer and Communication Networks, Springer, 2020, pp. 171–182.

[11] Á. Tóth, J. Sztrik, A. Kuki, T. Bérczes, D. Efrosinin: *Reliability analysis of finite-source retrial queues with outgoing calls using simulation*, in: 2019 International Conference on Information and Digital Technologies (IDT), IEEE, 2019, pp. 504–511.