

Erzsébet Csuhaj-Varjú and Péter Sziklai, editors

Conference

on

Developments in Computer Science

Budapest, Hungary, June 17-19, 2021

Proceedings

Faculty of Informatics, Eötvös Loránd University,
Budapest

Editors:

Erzsébet Csuhaj-Varjú

Institute of Computer Science

Faculty of Informatics

Eötvös Loránd University

Budapest, Hungary

E-mail: csuhaj@inf.elte.hu

Péter Sziklai

Institute of Mathematics

Faculty of Science

Eötvös Loránd University

Budapest, Hungary

E-mail: peter.sziklai@ttk.elte.hu

Conference on Developments in Computer Science,
Budapest, Hungary, June 17-19, 2021, Proceedings [PDF]

Published by Faculty of Informatics, Eötvös Loránd University,
1117 Budapest, Pázmány Péter sétány 1/c.

Edited by © Erzsébet Csuhaj-Varjú, Péter Sziklai, 2021

Copyright © Authors of the contributions, 2021

Published in December, 2021

ISBN 978-963-489-388-2

Preface

Erzsébet Csuhaj-Varjú¹ and Péter Sziklai²

¹Institute of Computer Science, Faculty of Informatics, Eötvös Loránd University ELTE,
Budapest, Hungary

²Institute of Mathematics, Faculty of Science, Eötvös Loránd University ELTE,
Budapest, Hungary

`csuhaj@inf.elte.hu`, `peter.sziklai@ttk.elte.hu`

This volume contains extended abstracts of talks presented at “Developments in Computer Science”, an online conference organized by the Faculty of Informatics and the Institute of Mathematics of the Faculty of Science, Eötvös Loránd University, Budapest, Hungary in the period 17 - 19 June, 2021.

The aim of the conference was to provide a forum for presenting current developments, ongoing works, inspiring ideas in all disciplinary areas of computer science. Researchers, lecturers working in these fields, as well as PhD and MSc students were encouraged to participate in the event and to exchange their ideas and results on the topics of the meeting.

The scope of the conference was broad and included topics both from theoretical fields of computer science and applications. The scientific program was organized in eleven sections, out of which ten sections were dedicated to specific research fields and one section was devoted to contributions on a few selected topics. The sections on the specific research fields consisted of an invited talk and several contributed talks.

In the section “Additive combinatorics and its applications in Computer Science”, organized by Norbert Hegyvári, an invited talk “Counting monochromatic solutions of the polynomial Schur equation $x + y = p(z)$ ” was presented by Péter Pál Pach.

Section “Coding theory and applications in cryptology” was organized by György Kiss, the invited presentation in this section was given by Marcella Takáts on “Secret sharing, coding theory and finite geometry”.

Section “Combinatorics and Geometry” was organized by Balázs Keszegh, the invited talk “Crossing lemma for the odd-crossing number and some related problems” was delivered by Géza Tóth.

In section “Geometric constraint systems: theory and algorithms”, organized by Tibor Jordán, the invited talk “Scene analysis with symmetry” was delivered by Viktória Kaszanitzky.

Section “Information Systems and Architectures” was organized by Bálint Molnár who also presented an introductory lecture “Formal approaches to modeling of Information Systems”.

Section “Neural networks and differential equations” was organized by Péter Simon, the invited talk “Adaptive numerical approximation of two-point boundary value problems: a neural network-based approach” was given by Ferenc Izsák.

In section “Numerical solution of differential equations, qualitative properties and applications”, organized by István Faragó, the invited speaker Róbert Horváth presented the invited talk with the same title, namely “Numerical solution of differential equations, qualitative properties and applications”.

Section “Type Theory” was organized by Ambrus Kaposi, the invited talk in this section, “Wellfounded and Extensional Ordinals in Homotopy Type Theory” was delivered by Nicolai Kraus.

In section “The ubiquitous machine learning – bridging science and business”, organized by András Lukács, the invited talk “Machine learning use cases in manufacturing” was presented by Szabolcs Biró.

Section “Using artificial intelligence tools in molecular structure prediction: The Budapest Amyloid Predictor and its applications”, was organized by Vince Grolmusz, and András Perczel presented the invited talk “The amyloid state of proteins” of the section.

Section “Selected topics” was organized by Erzsébet Csuhaaj-Varjú; the five regular contributions were presented on programming and computation theory.

The conference program started by the plenary talk “On the carriers of information (formations, infosphere, computation)”, delivered by András Benczúr.

The event was supported by the project “Integrált kutatói utánpótlás-képzési program az informatika és számítástudomány diszciplináris területein”, EFOP 3.6.3-VEKOP-16-2017-00002, a project co-financed by the Hungarian Government and the European Social Fund.

The Organizing Committee consisted of Erzsébet Csuhaaj-Varjú, Bálint Fügi, Hermina Molnár, Péter Sziklai and Nóra Tibold. The webpage of the conference was created and maintained by Gergő Gombos and Péter Vörös, who also contributed to the technical editing of this volume.

The editors thank the Program Committee, the section organizers, the invited speakers, the authors of the regular contributions, the reviewers, and all the participants who contributed to the success of the conference.

We thank the Faculty of Informatics and the Institute of Mathematics (Faculty of Science), Eötvös Loránd University for their support.

Budapest, July 2021

Erzsébet Csuhaaj-Varjú and Péter Sziklai

Contents

Preface	3
Extended Abstracts of Invited Talks and Regular Contributions	8
Plenary Talk:	9
András Benczúr: On the Carriers of Information (formations, infospher, computation)	11
Section:	
Additive Combinatorics and its applications in Computer Science	17
Péter Pál Pach: Counting monochromatic solutions of the "polynomial Schur equation" $x + y = p(z)$	19
Bence Bakos, Norbert Hegyvári and Máté Pálffy: A communication complexity problem; subset sums of Cartesian product of certain sets	23
Norbert Hegyvári: On a Boolean function defined on Number Theoretical structures	27
Richárd Palincza: The computational complexity of recognizing some number theoretic properties	31
Section:	
Coding theory and applications in cryptology	35
Marcella Takáts, Máté Gyarmati, Péter Ligeti and Péter Sziklai: Secret sharing, coding theory and finite geometry	37
Sabira El Khalfaoui and Gábor P. Nagy: Selecting secure parameters of Hermitian subfield subcodes for post-quantum schemes	41
Tamás Héger and Zoltán Lóránt Nagy: Short minimal codes and covering codes through geometric and probabilistic methods	47
Rebeka Kiss and Gábor P. Nagy: Correlation-immune Boolean functions and parameters of orthogonal arrays	51
Sára Pituk: MCF codes and multiple saturating sets	55
Section:	
Combinatorics and Geometry	59
János Karl and Géza Tóth: Crossing lemma for the odd-crossing number	61
Péter Ágoston: Semialgebraic sets as ranges of two-distance graphs	65
Gábor Damásdi and Nóra Frankl: A note on convex geometric hypergraphs	69
Gábor Damásdi, Dömötör Pálvölgyi: A generalization of the Erdős-Sands-Sauer-Woodrow conjecture	73
Rupert Levene and Narmada Varadarajan: Orthogonal Projections for Quantum Channels and Operator Systems	77
Section:	
Geometric constraint systems: theory and algorithms	81
Bill Jackson, Viktória E. Kaszanitzky and Bernd Schulze: Scene analysis with symmetry	83
Dániel Garamvölgyi: Algebraic matroids and global rigidity	87
Tibor Jordán: Rigid block and hole graphs with a single block	91

Csaba Király and András Mihálykó: Localizable sensor networks with optimal anchor sets I: A min-max theorem	95
Csaba Király and András Mihálykó: Localizable sensor networks with optimal anchor sets II: An algorithm	99
Section:	
Information Systems and Architectures	103
Bálint Molnár: Formal approaches for modelling Information Systems (Introductory talk)	105
Balázs Horváth and Bálint Molnár: Dynamic process modeling of micro-credentials	111
Meriem Kherbouche, Ahmad Mukashaty and Bálint Molnár: An Operationalized Transformation for Activity Diagram into YAWL	115
Zhang Yinghong and Bálint Molnár: An overview of reinforcement learning applications in the control system of the intelligent transportation system . .	125
Ekaterina Zolotareva, Bethelihem Seifu and Bálint Molnár: Credit risk management in financial services	129
Section:	
Neural networks and differential equations	133
Ferenc Izsák: Adaptive numerical approximation of two-point boundary value problems: a neural network-based approach	135
Petra Csomós, and Ferenc Izsák: In search of an appropriate loss function for differential equations' initial value problems	139
Domonkos Haffner and Ferenc Izsák: Solving the Laplace equation by using neural networks	143
Gábor Hidy: Residual neural networks as numerical approximations of differential equations	147
András Molnár, Imre Fekete, Péter L. Simon: Learning a function from data by solving a differential equation and tuning its parameters	151
Anita Windisch: Saddle-node bifurcation in a 3-dimensional neural network model	155
Section:	
Numerical solution of differential equations, qualitative properties and applications	159
Róbert Horváth: Numerical solution of differential equations, qualitative properties and applications	161
Teshome Bayleyegn and Ágnes Havasi: The method of multiple Richardson extrapolation	165
Lívía Boda: Operator splitting and Average method	169
Gabriella Svantnerné Sebestyén: Application of the Carleman linearisation method to partial differential equations	173
Bálint Takács, Róbert Horváth, István Faragó and Yiannis Hadjimichael: Numerical methods for space-dependent epidemic models	177
Section:	
The ubiquitous machine learning – bridging science and business .	181
Szabolcs Biró and Szilárd Varró: Machine Learning Use Cases in Manufacturing	183
Bálint Csanády and András Lukács: 1D Convolutional Neural Networks for Diacritics Restoration	187
Gábor Hidy and András Lukács: Nucleus classification with neural networks . . .	191

Gellért Károlyi and András Lukács: Transfer learning for medical image classification 195

Melinda Kiss, Adrián Csiszárík, Ákos Matszangosz, Balázs Maga, and Dániel Varga: Global Sinkhorn Autoencoder - Optimal transport on the latent representation of the full dataset 199

Péter Marton, Norbert Bicskei, András Lukács: Machine Learning Algorithms for MOD Lapse at renewal 203

Section:

Type Theory 207

Nicolai Kraus: Wellfounded and Extensional Ordinals in Homotopy Type Theory (talk on joint work with Fredrik Nordvall Forsberg and Chuangjie Xu) 209

István Donkó and Ambrus Kaposi: Properties of Setoid Type Theory 213

Ambrus Kaposi and Zongpu Xie: A model of type theory supporting quotient inductive-inductive types 217

András Kovács: Staged Compilation and Generativity 223

Section:

Using artificial intelligence tools in molecular structure prediction: The Budapest Amyloid Predictor and its applications 229

András Perczel: The amyloid state of proteins 231

László Keresztes: Amyloid patterns in hexapeptides 233

Evelin Szögi: Predicting the amyloid state by Support Vector Machines 237

Kristóf Takács and Vince Grolmusz: Sliding windows in the Protein Data Bank: amyloid-forming propensity of prefixes and suffixes of secondary structures 241

Bálint Varga: Pathfinding in the hexapeptide-graph: through the amyloid and non-amyloid nodes 245

Section:

Selected Topics 249

Zsófia Erdei, Melinda Tóth and István Bozó: Targeted static fault localization in Erlang programs 251

Beka Grdzlishvili and Viktória Zsók: Design and Implementation of Digital Image Processing in Functional Programming 255

Jianhao Li and Viktória Zsók: Actor Model based Distributed Communication in Golang 259

Pramod Kumar Sethy: Notes on P systems versus R systems 263

Gabriella Tóth and Máté Tejfel: Error detection and analysis of PSA structured P4 programs 267

Plenary Talk:
On the carriers of information
(formations, infosphere, computation)

by **András Benczúr**



On the Carriers of Information *(formations, infospher, computation)*

András Benczúr

Faculty of Informatics, Eötvös Loránd University
Budapest, Hungary
abenczur@inf.elte.hu

In my presentation, I present the world of information from a specific, new perspective. I do not attempt to define the concept of information itself, but I base my analysis on the examination of the carriers of information. The carriers of information are referred to as formation in the following in order to distinguish and yet refer to information. Formations are objects that physically exist for shorter, longer periods of time. They occur both in human brains and as artificial external objects. Artificially created external formations elevated man to the top of conscious (nervous) beings. Formations are linked to their meaning during information events. The rich set of formations that can be used in our time was made possible by the mutually supportive development of our two human abilities: the ability to rearrange matter and the ability to think and remember. In the possibilities of rearranging the material, we have now reached the point where – although we cannot make the material do thinking, but - we can make it perform calculation. Rearranging matter and thinking come together: some activities of thinking can be done with matter itself. What is doable can be boldly called computation. The border between thinking and calculating is yet to be clarified, has provoked much controversy so far, and is expected to provoke more.

1 Formations and information, information event, the info-sphere and the information revolution

We briefly review the most important levels of material rearrangement in the evolution of the world of formations and information. The first three levels, the first, the initial rearrangement of material in creation of tools and cultural objects, the second, the thinking and remembering and the third, the pronouncing of a sequence of separable sounds into sound formations, led to the development of words and language. Words are perceptible formations and can be considered as encodings with a finite set of symbols. Writing is the next level, followed by printing, then by a microscope and telescope that refine visual perceptions. The creation of external formations up to this level required human activity, the printing only multiplied the formations.

During the last two centuries developed and still is developing the layer of instrumental perceptions and recordings, in which formations are made about the perceptions, that can be stored and transmitted. They can then be processed with the means of the next layer.

Last layer, the determinant of our era, brings in the rearrangement of material so that it rearranges itself to perform the computation. A new artificial active layer is evolving

in exponential pace. The special feature of the new artifact, the computer (computing machinery), is that the material it transforms is “only” a formation. This also means that it becomes useful and becomes information only in the case of the appropriate meaning assigned to the formation. The solution of the assignment is the challenge and task of informatics.

In the era of the information society and the information revolution, information has got central role, and the use of the word information became commonplace. However, there is no accepted unifying definition of information. A model is needed where information takes meaning from the formation, where the representative of the information and the meaning itself are present at the same time. We can assume that information does not exist without material form, it is carried by something, that can be observed / perceived / shaped. This carrier is what I have already called a formation. When the formation is observed / created, then the information appears to the observer / creator and the meaning is associated to the formation. Based on the introductory preparation, I give a new definition consisting of a combination of three components.

Definition 1 *The information triad is the triple of formation, owner, and meaning. The role of formation in the triad can be of three kinds: generated, perceived, or emitted. This triple belongs to a process, happening in time, that I call an information event. Information in this definition is a timely presence of a formation that the owner can relate to meaning, to a referenced. The owner either associates a new formation with the referenced, or detects a formation associated with his referenced, or issues a formation as a replacement for the referenced.*

I give two basic laws:

The **first basic law**: Different meanings require different formations (distinguishable by the observer / creator).

The **second basic law**: Formation detection / creation means information if at the same time it is possible to detect previously formed formations to which the meaning is linked.

The second basic law introduces the time dependence of information and the quality of understanding and utilization. Semantic modeling should take this into account as well. This applies to information events in both the human and artificial spheres.

The most important information events are related to activity of human consciousness. The human mind is the owner. Formations can be internal structures of the brain (internal-formations) and can be external, materialized structures (external-formations). Brain formations within the consciousness should also be perceived, and conscious movements can also be considered referenced. Such is our mental inner world, our emotions, our thoughts. The length of the information event is not limited, complex formations are possible, several owners can participate in it at the same time. Persistent or repeatable formations allow for communities to assign meaning to them with consensus. The accumulation of such formations provides the community with a collection of information. The information events of the artificial world are based on preceding human information events.

By focusing on material rearrangement in the introduction, I wanted to support that the accumulation of information is embodied in physically existing media, which means collections of formations. What does the current stock consist of? From the individual sets in the consciousness of the living people and from the stock of objectified, external formations. More and more advanced systems and information technologies have been developed for accessing, processing, distributing and operating these stocks. Together with the collection of formations, they form the infosphere. However, formations only become information during an information event, so the meaning ultimately always appears in human consciousness.

Construction based on formations and information events does not define the concept of information. It captures in what form and during what event the information is present. Further definition and classification of information can refer to the specific sets and structures of the formations used / usable, to the situations of information events and to the types of phenomena represented. Alternatively, we can approach some specificity of information from all three components. After a general presentation of formations, information events, and the accumulation of information, I turn to characterizing the new information world of our era; the new infosphere. The infosphere, like the biosphere, is extremely rich and diverse. The digital universe is currently at the peak of this development. The entire infosphere is brought to life at all times by the information carriers of the individual (living consciousness). What is the essence of the new information revolution? Before the world of computers, the external representations and fixed forms of information were passive, they did not undergo transformation, at most they wore out, deteriorated. They could trigger activity and they could be used to produce additional information only after human perception. However, the formations that have entered the data world of computers can be transformed during the execution of implemented algorithms and programs without human intervention, and can also influence the operation of artificial objects. This will lead to a new revolutionary development of the infosphere, including the construction of the digital universe and the connection of more and more elements of everyday life.

2 Formations, computation and informatics

The utilization, use, transformation and processing of information has long been only through human consciousness. This field has been revolutionarily changed by the computing machines and the digital world. In Part 2 the strong interrelationship of computation and information is dealt with. The novelty of the presentation is given by the analysis based on the formations and the information events.

There are two kinds of formations in the infosphere: artificial outer/ex-formations and natural inner/in-formations. The permanent task to be solved is: how to get information from the collections of artificial physical formations? In solving the problem, two periods can be distinguished: the period before the computation and the period after the computation. (Before Computation and After Computation).

In the BC era the basic solution is: one has to find and access the carriers (clay and paper) of relevant set of formations, observe (read), understand and extract the necessary information.

There is a daily flood of formation that reaches everyone. The most important category is the set of formations not intended for immediate consumption or communication. These formations are carried by physical medium that have been physically existing from the time they were created. These are collected in formation repositories. (Libraries, document collections, records, archives, etc.) Supporting access to the formation relevant to a question additional formations are necessary for access and content relationships (catalogs, thesaurus, indexes). These are also added to the set of formation carriers. This is what LIS (Library and Information Science) is all about.

In the post-computational (AC) era, we have come to the point where overwhelming and growing proportion of external formations exist digitally somewhere in computer systems connected to the global Net. The Digital Universe has been built, with the data sphere representing the data world, according to the latest name of the IDC study. IDC has defined three primary locations where digitization is happening and where digital content is created: the core (traditional and cloud datacenters), the edge (enterprise-hardened infrastructure like cell towers and branch offices), and the endpoints (PCs, smart phones, and IoT devices). The summation of all this data, whether it is created, captured, or replicated, is called the Global Datasphere, and it is experiencing tremendous growth. IDC predicts that the Global Datasphere will grow from 33 Zettabytes (ZB) in 2018 to 175 ZB by 2025.

Capabilities of computation, which was well-known and used for ages, extended over digital signs, furthermore computation can be carried out by machines. This is the point where the two components of information (Formation, Referent) can be connected to the computation. Suppose, that some other properties of the represented entity can be inferred from the available information. As the result of formal reasoning we can get a valid representation of not observed properties. It was done previously only by mental processes - nowadays, by machine computation. However, we can obtain such a result for a problem which have computational model that was created by us. Thinking and computation are processes developing in space and time, they work with formations, typically observable at the beginning and at the end. As physically existing events, they are carrying new observable information. Extracting meaning is the duty of the observer. This process is the resolution of Denning' and Bell's Paradox: *How can machines that work independently of meaning generate meaning for observers? Where does the new information come from?*

Thinking and computation are processes that unfold in space and time, with formations that are typically observed at the beginning and end. They carry new information as physically existing events. As a new observation of the resulting formation, the owner can associate the meaning to it, so this is new information. The novelty of this information usually depends on the complexity of the calculation.

P. Denning gave a definition for the computation with representations. I rephrase this using formations instead of representations, which brings computing closer to the world of information.

I make some note on the mathematical definition of computation, what is based on the Turing-machine and the Church-Turing thesis. Whether the axiom of computation can

be based on a Turing-machine is disputed by others, supported by examples of interactive algorithms, continuously operating on-line algorithms, distributed systems. New computational capabilities, such as parallel, interactive, continuous computations, do not correspond to the uninterrupted operation of a Turing-machine, but they can be implemented by using Turing-automata for non-exclusive tape use. In this sense, I extend the CT-thesis to Turing-automata.

A short description is given of the main models of mathematical information theory from the perspective of efficient management of formations and information events. I point out that the metrics of information (Shannon entropy, Kolmogorov entropy) are related to the choice of formation sets and systems, with their ability to briefly describe the sets and structures of the possible referents. I discuss the role of the universal Turing machine in algorithmic information theory. I present the phenomenon of the growing data gap associated with the growth of the data sphere by modeling the query process between man and computer.

3 Summary

In my presentation, starting from the human possibilities of material rearrangement and thinking, I directed attention to the material manifestations of information. The introduction of formations can make visible the boundary between the information construction of humanity and the complex processes of nature. The set of formations that physically exist at any time and the set of systems that ensure its use make up the infosphere of humanity. The infosphere is a product of the biosphere, it would not exist without it, since there would be no human being outside the biosphere. And today's age represents a new evolutionary step in building the infosphere with the new digitization and mechanization of computing.

All this leads to the emergence of new science. The new development of the infosphere already requires its own field of science, it can be informatics, and as a discipline Informatical Sciences, similarly to mathematics and Mathematical Sciences.

References

- [1] Benczúr, A.: On the Carriers of Information: Information from a New Perspective, Part 1. On the Relationship of Information, Formations, and the Infosphere; Magyar Tudomány, 2021.June. Part 2. On the Relationship of Information, Computation and Informatics; Magyar Tudomány. 2021. July.
- [2] Benczúr, A.: On the Carriers of Information (formations, infospher, computation). Presentation, July 2021. DOI: 10.13140/RG.2.2.22045.84966



Section:
**Additive Combinatorics and its applications in Computer
Science**

Organizer: Norbert Hegyvári

Invited talk:

Péter Pál Pach: Counting monochromatic solutions of the "polynomial Schur equation" $x + y = p(z)$

Contributions:

- Bence Bakos, Norbert Hegyvári and Máté Pálffy: A communication complexity problem; subset sums of Cartesian product of certain sets
- Norbert Hegyvári: On a Boolean function defined on Number Theoretical structures
- Richárd Palincza: The computational complexity of recognizing some number theoretic properties



Counting monochromatic solutions of the “polynomial Schur equation” $x + y = p(z)$

Péter Pál Pach

MTA-BME Lendület Arithmetic Combinatorics Research Group,
Department of Computer Science and Information Theory,
Budapest University of Technology and Economics,
1117 Budapest, Magyar tudósok körútja 2., Hungary
ppp@cs.bme.hu

Abstract

We discuss the Ramsey problem for $\{x, y, z : x + y = p(z)\}$ for all polynomials p over \mathbb{Z} .

Under the assumption that $p(1)p(2)$ is even we show that $x + y = p(z)$ is 2-Ramsey. Indeed, we show that the number of monochromatic solutions with $x, y, z \in \{1, 2, \dots, n\}$ is at least $n^{2/d^3 - o(1)}$, where $d = \deg p$. On the other hand, there exists a 2-colouring for which the number of monochromatic solutions is at most n^{2/d^2} . Furthermore, in *almost all* of the cases we can improve the lower bound to $n^{2/d^2 - o(1)}$.

On the other hand, when $p(1)p(2)$ is odd, that is, when p attains only odd values, then there might not be any monochromatic solution, for instance, this is the case when we colour the integers according to their parity. We give a characterization of all 2-colourings avoiding monochromatic solutions to $x + y = p(z)$.

The talk is based on joint work with Liu and Sándor and with Kim and Liu.

1 Introduction

The study of arithmetic Ramsey theory searches for monochromatic patterns in finite colourings of \mathbb{N} . A pattern is *k-Ramsey*, if it appears *infinitely* often in any *k*-colouring of \mathbb{N} . Ramsey theory has a long history dating back to the famous theorem of Schur in 1916, which states that the equation $x + y = z$ is Ramsey, that is, any finite colouring of \mathbb{N} contains infinitely many monochromatic solutions to $x + y = z$. Note that it is wide open whether the Pythagorean equation $x^2 + y^2 = z^2$ is also Ramsey. Heule, Kullmann and Marek [4] gave a computer-assisted proof (of size 200 terabytes!) that $x^2 + y^2 = z^2$ is 2-Ramsey, in fact any 2-colouring of $\{1, 2, \dots, 7825\}$ admits a monochromatic solution, while $\{1, 2, \dots, 7824\}$ can be 2-coloured avoiding monochromatic solutions. However, it is still open whether each 3-colouring of \mathbb{N} admits a monochromatic solution.

Another classical example is van der Waerden’s theorem [7] stating that $\{x, x + y, \dots, x + (\ell - 1)y\}$ is Ramsey for any $\ell \in \mathbb{N}$. Rado [6] later in his seminal work resolved the Ramsey problem for all *linear* equations, characterising all those that are Ramsey. Since then, many extensions have been studied, see e.g. the far-reaching polynomial extension of van der Waerden’s theorem by Bergelson and Leibman [1].

In this talk, we discuss the polynomial extension of Schur’s theorem. Somewhat surprisingly, only a special case of this natural problem has been solved. Csikvári, Gyarmati and Sárközy [2] showed that $x + y = z^2$ is *not* 16-Ramsey, that is, they constructed a

16-colouring of \mathbb{N} with no monochromatic solution to $x + y = z^2$ other than the trivial solution $x = y = z = 2$. Later, Green and Lindqvist [3] completely resolved this case using Fourier-analytic arguments, giving the satisfying answer that any 2-colouring of \mathbb{N} contains *infinitely* many monochromatic solutions, while 3 colours suffice to avoid non-trivial monochromatic solutions. In other words, $x + y = z^2$ is 2-Ramsey, but not 3-Ramsey. In fact, the 3-colouring in [3] can easily be adapted to show that

$$x + y = p(z) \text{ is not 3-Ramsey for any } p(z) \in \mathbb{Z}[z] \text{ with } \deg(p) \geq 2.$$

The result in [3] also implies that there are at least $\log \log N$ monochromatic solutions in $[N] := \{1, \dots, N\}$ for any sufficiently large N . On the other hand, there is a greedy 2-colouring with at most $N^{1/2}$ monochromatic solutions.

The Fourier-analytic proof in [3] actually shows that for any sufficiently large N , there is a monochromatic solution in the interval $[N, cN^8]$ for some large constant c .

Recently, we [5] gave a shorter combinatorial proof for the 2-Ramsey of $x + y = z^2$, showing that there is a monochromatic solution to $x + y = z^2$ in the smaller interval $[N, 10^4 N^4]$, and the bound on the interval is optimal up to the constant factor.

Here, we completely resolve the Ramsey problem for

$$\{x, y, z : x + y = p(z)\}$$

for all polynomials p over \mathbb{Z} , thereby establishing a polynomial extension of Schur's theorem. In particular, we characterise all polynomials that are 2-Ramsey.

2 Results

For polynomials that are 2-Ramsey, we have a quantitative result, giving a lower bound of the correct shape on the number of monochromatic solutions. Note that the condition $a_d > 0$ is necessary as otherwise $p(z)$ would eventually take only negative values. The assumption $2 \mid p(1)p(2)$ is also needed, since otherwise $p(z) \equiv p(1)p(2) \equiv 1 \pmod{2}$ and one can 2-colour \mathbb{N} by parities to avoid monochromatic solutions.

Theorem 1 (H. Liu, C. Sándor, P. P. Pach) *Let $p(z) = a_d z^d + \dots + a_1 z + a_0 \in \mathbb{Z}[z]$ with $d \geq 1$ and $a_d > 0$ such that $2 \mid p(1)p(2)$. Let ϕ be a 2-colouring of $[n]$. Then the number of monochromatic solutions $\{x, y, z\} \in [n]^{(3)}$ to $x + y = p(z)$ is at least $n^{2/d^3 - o(1)}$. Moreover, there is a 2-colouring for which the number of monochromatic solutions is only $O(n^{2/d^2})$.*

In ongoing work with Kim and Liu we could prove that in fact $n^{2/d^2 - o(1)}$ is also a lower bound for the number of monochromatic solutions when $d = 2$ or $d \geq 4$, that is, only the degree-3 case remains open.

On the other hand, for polynomials that are not 2-Ramsey, we characterise all 2-colourings of \mathbb{N} that are not 2-Ramsey, showing that all such *bad* 2-colourings have to be *balanced* and *periodic*. Moreover the sumset of each colour class must have a rigid structure. It further reveals that a *divisibility* barrier, generalising the aforementioned parity obstruction, is the *only* obstruction to 2-Ramsey for $x + y = p(z)$.

Theorem 2 (H. Liu, C. Sándor, P. P. Pach) *Let $p(z) = a_d z^d + \dots + a_1 z + a_0 \in \mathbb{Z}[z]$, with $d \geq 1$ and $a_d > 0$. Let $\phi : \mathbb{N} \rightarrow \{-1, 1\}$ be a 2-colouring such that $x + y = p(z)$ does not have infinitely many monochromatic solutions. Then there exists an even positive integer m and a partition of \mathbb{Z}_m into two classes A and B , each of size $m/2$, such that*

$$\phi(x) = -1 \quad \text{if and only if} \quad x \in A \pmod{m}.$$

Furthermore, there exists an odd $\alpha \in \mathbb{Z}_m$ such that

$$A + A = B + B = \mathbb{Z}_m \setminus \{\alpha\},$$

and for any $z \in \mathbb{N}$, we have

$$p(z) \equiv \alpha \pmod{m}.$$

Note that if ϕ and p satisfies the above conditions, then $p(z) \equiv \alpha \pmod{m}$ for every z , however, whenever x and y have the same colour $x + y \not\equiv \alpha \pmod{m}$. Thus there is no monochromatic solution, even trivial ones do not exist. In other words, if $x + y = p(z)$ has a trivial solution, such as $x = y = z = 2$ for $x + y = z^2$, then the polynomial p is necessarily 2-Ramsey. Therefore, the following corollary is obtained.

Collorary 3 *Let $p(z) = a_d z^d + \dots + a_1 z + a_0 \in \mathbb{Z}[z]$ with $d \geq 1$ and $a_d > 0$ and ϕ be a 2-colouring of \mathbb{N} . Either there is no monochromatic solution for $x + y = p(z)$, or there are infinitely many monochromatic solutions.*

A special case of the periodic colouring is the one induced by parity and a polynomial for which $p(1)p(2)$ is always odd. Below is another example illustrating the divisibility barrier to 2-Ramseyness for $x + y = p(z)$.

Example. Consider

$$p(z) = z^3 + 3z^2 + 2z + 3 = z(z + 1)(z + 2) + 3.$$

Note that for every $z \in \mathbb{N}$,

$$p(z) \equiv 3 \pmod{6}.$$

Colour all numbers that are 2, 3, 5 modulo 6 with one colour, and the rest, 0, 1, 4 modulo 6, with the other colour. One can easily check that any number having residue 3 (mod 6) cannot be written as a sum of two numbers of the same colour.

3 A brief overview of the methods

We present in this section the proof sketch for Theorem 2: characterising all pairs of polynomials p and 2-colourings ϕ such that $x + y = p(z)$ does not have any (or equivalently, does not have infinitely many) ϕ -monochromatic solutions (Theorem 2). For the lower bound on the number of monochromatic solutions in $[n]$ (Theorem 1) similar methods are used, however, additional difficulties need to be overcome.

Trivially, if there is a "very long" monochromatic interval, then many monochromatic solutions can be found in it. Thus, we may assume that there will be infinitely many places where the colour switches. With the help of a simple, but crucial observation we

can see that whenever a “sufficiently long” block of numbers of one colour is followed by a sufficiently long block of numbers coloured with the other colour, many monochromatic solutions can be found. This allows us to assume that the colour switches “frequently” after some threshold.

When considering a switch k , that is, $\phi(k) \neq \phi(k+1)$, we define a subset $A = A_k \subseteq \mathbb{Z}_{m(k)}$ (where $m = m(k) := p(k+1) - p(k)$) containing at least half of the elements of \mathbb{Z}_m . The set A satisfies that whenever $z \in \mathbb{N}$ is such that (i) $p(z)$ lies in the sumset $A + A \pmod{m}$, and (ii) z has the opposite colour of k , then we are able to find a monochromatic solution. To drop restriction (ii) on the colour of z , we shall use that the colour switches frequently, according to the above discussion. If k_1 and k_2 are two consecutive switches, then clearly $\phi(k_1) = -\phi(k_2)$ and either k_1 or k_2 would have the opposite colour of z .

However, we still need to guarantee (i) that $p(z) \in A + A \pmod{m}$. As $A \subseteq \mathbb{Z}_m$ contains at least $m/2$ elements, by the pigeon-hole principle $A + A = \mathbb{Z}_m$ holds if $|A| > m/2$, and then $p(z) \in A + A$ is automatically satisfied. If $|A| = m/2$, then the sumset $A + A$ might not contain all elements of \mathbb{Z}_m . These cases can be described with the help of a stability version of the Cauchy-Davenport theorem) and indeed the union of the residue classes outside of the sumset $A + A$ form a residue class α modulo m' for some even $m'|m$.

Now, if we obtain the same α and m' infinitely often, then this forces the periodic structure of the colouring and $p(z) \equiv \alpha \pmod{m'}$ for all z . Otherwise we would get a sequence $m' \rightarrow \infty$. However, for a fixed polynomial p it is not possible to have $p(z) \equiv \alpha \pmod{m'}$ for all z if m' is sufficiently large. More precisely, with the help of Szemerédi’s theorem on arithmetic progressions, we prove this for a pair of moduli m'_1, m'_2 , as to drop the condition on $\phi(z)$ we work with pairs of switches.

References

- [1] Bergelson, V., Leibman, A., Polynomial extensions of van der Waerden’s and Szemerédi’s theorems, *J. Amer. Math. Soc.*, **9** (3) (1996), 725–753.
- [2] Csikvári, P., Gyarmati, K., Sárközy, A., Density and Ramsey type results on algebraic equations with restricted solution sets, *Combinatorica*, **32** (2012), 425–449.
- [3] Green, B., Lindqvist, S., Monochromatic solutions to $x+y = z^2$, *Canadian Journal of Mathematics*, **71** (3) (2019), 579–605.
- [4] Heule, M., Kullmann, O., Marek, V. W., Solving and Verifying the Boolean Pythagorean Triples problem via Cube-and-Conquer, In Theory and Applications of Satisfiability Testing – SAT 2016, pp. 228–245.
- [5] Pach, P. P., Monochromatic solutions to $x + y = z^2$ in the interval $[N, cN^4]$, *Bulletin of the London Mathematical Society*, **50** (6) (2018), 1113–1116.
- [6] Rado, R., Studien zur Kombinatorik, *Math. Z.*, **36** (1933), 424–470.
- [7] van der Waerden, B. L. Beweis einer baudetschen vermutung, *Nieuw. Arch. Wisk.*, **15** (1927), 212–216.

A communication complexity problem; subset sums of Cartesian product of certain sets

Bence Bakos, Norbert Hegyvári and Máté Pálffy

Eötvös University, Institute of Mathematics,
1117 Budapest, Pázmány st. 1/c, Hungary

{bakosbence237,n.hegyvari,palfymateandras}@gmail.com

1 Introduction and Motivation

In the last decades there were several interplay between computer sciences and additive combinatorics. One of the most interesting example is an additive communication complexity problem which was supported by an example of Behrend on the maximal density of a set without a three-term arithmetic progression (see e.g. [7]) We investigate a communication complexity problem which is related to a field in combinatorial number theory; namely to the topic of subset-sums. The well-known knapsack problem is to determine, given positive integers y_1, y_2, \dots, y_n and an integer n , whether there is a subset of the set $\{y_j\}$ that sums up to N . This problem belongs to the class of NP-complete problems. Certainly this question can be extend to higher dimension too. For any $X \subseteq \mathbb{N}^k$ let

$$FS(X) := \left\{ \sum_{i=1}^{\infty} \varepsilon_i x_i : x_i \in X, \varepsilon_i \in \{0, 1\}, \sum_{i=1}^{\infty} \varepsilon_i < \infty \right\} \quad (1)$$

and we have to decide for $p \in \mathbb{N}^k$ whether $p \in FS(X)$ or not.

Let $X = A_1 \times A_2 \times \dots \times A_k \subseteq \mathbb{N}^k$. The structure of $FS(X)$ in higher dimension was investigated in [1], [2] and [3].

The set $Y \subseteq \mathbb{N}$ is said to be *complete* if $FS(Y) = \mathbb{N}$. Let us note that although $FS(A_i) = \mathbb{N}$ then $FS(X)$ does not cover necessary the whole \mathbb{N}^k . For example if $X = \{2^k\}_{k=0}^{\infty} \times \{2^m\}_{m=0}^{\infty}$ then $(15, 1) \notin FS(X)$, while $(15, 1 + 256) \in FS(X)$.

2 A communication complexity problem

we will use number-in-hand multiparty communication model, i.e. there are k players P_1, P_2, \dots, P_k and a k -argument functions $F : (\{0, 1\}^N)^k \mapsto \{0, 1\}$. For every $i \in [k]$ P_i gets an n -bit input. In the communication process we will use *blackboard model* where every message sent by a player is written down on a blackboard which is visible for all players.

The communication complexity of this model, denoted by $CC^{(k)}(F)$, is the least number of bits needed to be communicated to compute F correctly.

Assume that we have k players and we assign a regular sequence A_i to each of them. For a given point $p = (p_1, p_2, \dots, p_k) \in \mathbb{N}^k$; $p_i \leq N$; ($i = 1, 2, \dots, k$), the i^{th} players knows (just) p_i and his previously given set A_i . Let $X := A_1 \times \dots \times A_k$. With minimal communications they have to decide whether $p \in FS(X)$ or not. Denote by F the function which describes this.

3 Results

As we indicated X is a cartesian product of certain sets. Namely $X := A_1 \times \dots \times A_k$, where for every $i = 1, 2, \dots, k$

$$(i) 1 \in A_i; \quad (ii) A_i \setminus \{1\} \subseteq A_i + A_i; \quad (iii) a_{j+1} > \varrho a_j$$

In the next we will show that sequences which fulfill conditions (i) and (ii) are complete.

Proposition 1 *Let $Y \subseteq \mathbb{N}$ be an infinite set and assume that $1 \in Y$ and $Y \setminus \{1\} \subseteq Y + Y$. Then Y is complete. Moreover if $Y = \{1 = y_1 < y_2 < \dots\}$ then for every $i = 1, 2, \dots$ we have $y_i \leq 2^{i-1}$.*

3.1 Regular sequences

For any $z \in \mathbb{N}$ and $X \subseteq \mathbb{N}$ let us denote by $r(z)$ the representations of z from X , i.e. $r(z) := r_X(z) = \{(\eta = \{\eta_i\}_{i=1}^\infty) : z = \sum_{i=1}^\infty \eta_i x_i, x_i \in X, \sum_{i=1}^\infty \eta_i < \infty, \eta_i \in \mathbb{N}\}$. Note that in this form it is allowed to use an element of X more than once. We are going to look at shortest representations of $n \in \mathbb{N}$ and for this we will use the notation $rank_Y(p) := \min_{\eta \in r_Y(p)} (\sum_i \eta_i)$, i.e. the length of the shortest representation of p from the set Y . Denote by $mult_Y(p, \eta)$ the maximal multiplicity of an element in the representation ε from $r_Y(p)$. For example if $p = a_1 + a_2 + a_2 + a_3 + a_3 + a_3$ ($\eta_1 = 1, \eta_2 = 2, \eta_3 = 3$) then $mult_A(p, \eta) = 3$

A sequence A is said to be *regular* if all numbers n have a shortest representation which has multiplicity equal to 1.

Note that many 'classical' sequences which fulfill conditions (i),(ii), and (iii) (e.g. the sequence of two powers, the Fibonacci sequence) are also regular.

Lemma 2 *The sequences of two powers and the Fibonacci numbers ($F_1 = 1, F_2 = 2, \dots$) are regular and fulfills conditions (i)-(iii).*

Main result on regular sequences. Recall that the communication complexity of this model, denoted by $CC^{(k)}(F)$, is the least number of bits needed to be communicated to compute F correctly.

Assume that we have k players and we assign a regular sequence A_i to each of them. For a given point $p = (p_1, p_2, \dots, p_k) \in \mathbb{N}^k$; $p_i \leq N$; ($i = 1, 2, \dots, k$), the i^{th} players knows (just) p_i and his previously given set A_i . Let $X := A_1 \times \dots \times A_k$. With minimal communications they have to decide whether $p \in FS(X)$ or not. Denote by F the function which describes this.

Our main result is the following:

Theorem 3 *Let $X = A_1 \times A_2 \times \dots \times A_k \subseteq \mathbb{N}^k$, where for every $i = 1, 2, \dots, k$ A_i is regular and (i), (ii) and (iii) hold. Then*

$$CC^{(k)}(F) < k \log_2 \left(\frac{\log_2 N}{\log_2 \varrho} \right) + k.$$

3.2 On non-regular case

Let $X = A_1 \times A_2 \subseteq \mathbb{N}^2$. As we have seen there are sequences, where the shortest representation of an element of $FS(A_i)$ ($i = 1, 2$) is not a subset-sum. It is not obvious that which conditions ensure that a given point (p_1, p_2) is an element of $FS(X)$ or not. Clearly it is necessary to have (multi)partitions of $p_1 = y_{t_1} + y_{t_2} + \dots + y_{t_r}$ and $p_2 = y_{i_1} + y_{i_2} + \dots + y_{i_k}$ for which $r = k$. One can show that it is not sufficient.

Nevertheless if $X = A_1 \times A_2$, where A_1 and A_2 fulfill conditions (i), (ii) and (iii), we will show an additional condition which is enough.

Theorem 4 *Let A_1 and A_2 fulfill conditions (i), (ii) and (iii) and let $X = A_1 \times A_2$. Let $p_1, p_2 \in \mathbb{N}$ with $\text{rank}_{A_1}(p_1) = \text{rank}(p_1)$ and $\text{rank}_{A_2}(p_2) = \text{rank}(p_2)$ (for simplicity), and let ε_1 and ε_2 be shortest representations of p_1 and p_2 . Write $M_i := \text{mult}(p_i, \varepsilon_i)$; $i = 1, 2$. Let $K := \max\{M_1, M_2, |\text{rank}(p_1) - \text{rank}(p_2)|\}$ and $L := \min\{\text{rank}(p_1), \text{rank}(p_2)\}$.*

If there are ε_1 and ε_2 for p_1 and p_2 such that, $K \leq \sqrt{L}/2$, then $(p_1, p_2) \in FS(X)$.

For the proof the main tool is the following graph theoretical lemma which gives a sufficient condition for the pairwise different matching of the co-ordinates:

Lemma 5 ([1, Proposition 1]) *Let X_1, \dots, X_s be disjoint finite sets and Y_1, \dots, Y_t be disjoint finite sets too. Let*

$$U = \bigcup_{i=1}^s X_i, \quad V = \bigcup_{j=1}^t Y_j,$$

with $|U| = |V|$ and suppose that $1 \leq |X_i| \leq \sqrt{|U|}$ for $i = 1, 2, \dots, s$ and $1 \leq |Y_j| \leq \sqrt{|V|}$ for $j = 1, 2, \dots, t$. Then there exists a bipartite graph $G(U, V)$ fulfilling the following conditions:

- (a) *there are no two edges $(x_1, y_1); (x_2, y_2)$ for which $x_1, x_2 \in X_i$; $y_1, y_2 \in Y_j$ for some i and j ;*
- (b) *$G(U, V)$ is a matching.*

4 Concluding remarks

Recall that the general knapsack problem is known to be NP-complete and sounds as follows: for a given sequence $A = \{a_1, a_2, \dots, a_n\} \subset \mathbb{N}$ decide that the equation $s = \sum_{i=1}^n \varepsilon_i a_i$; $\varepsilon_i \in \{0, 1\}$, $i = 1, 2, \dots, n$ is solvable or not in $\varepsilon_1, \varepsilon_2, \dots, \varepsilon_n$.

The density of a knapsack problem is defined as:

$$d := \frac{n}{\log_2(\max a_i)}.$$

When $d < 1$ then there is a possible encryption process. When $d > 1$ there is no an effective approach to attack the knapsack problem. The main tool is the so called basis reduction method.

Now we will show a way to reduce this problem, decide whether a given point (p_1, p_2) is an element of $FS(A_1 \times A_2)$ or not, to a classical knapsack problem.

Let $(p_1, p_2) \in \mathbb{N}^2$ and assume that $1 \leq p_1, p_2 \leq M$. Let $B_1 := \{x_1 < x_2 < \dots < x_k\} \subset A_1$ and $B_2 := \{y_1 < y_2 < \dots < y_m\} \subset A_m$, where $k = \max\{T : x_1 + x_2 + \dots + x_T \leq M\}$ and $m = \max\{R : y_1 + y_2 + \dots + y_R \leq M\}$.

Let now $Z := \{z = Mx_i + y_j : 1 \leq i \leq k; 1 \leq j \leq m\} \subset \mathbb{N}$. Observe that $(p_1, p_2) \in FS(A_1 \times A_2)$ if and only if $Mp_1 + p_2 \in FS(Z)$.

Acknowledgement. This work is supported by the European Union, co-financed by the European Social Fund (EFOP-3.6.3-VEKOP-16-2017-00002).

References

- [1] Chen, YG ; Fang, JH ; Hegyvári, N Erdős-Birch type question in \mathbb{N}^r J. of Number Theory 187 pp. 233-249. , 17 p. (2018)
- [EG80] P. Erdős, R. L. Graham: Old and new problems and results in combinatorial number theory: van der Waerden's theorem and related topics, Enseign. Math. (2) 25 (1979) no. 3–4, 325–344 (MR81f:10005)
- [3] Hegyvári, N Subset sums in \mathbb{N}^2 Combinatorics Probability and Computing (5) (1996) 393-402.
- [7] A. Rao, A. Yehudayoff, Communication Complexity, available at <https://homes.cs.washington.edu/~anuprao/pubs/book.pdf>

On a Boolean function defined on Number Theoretical structures

Norbert Hegyvári

Eötvös University, Institute of Mathematics,
and Alfréd Rényi Institute of Mathematics, Hungarian Academy of Science
1117 Budapest, Pázmány st. 1/c, Hungary
hegyvari@renyi.hu

1 Introduction and Motivation

A function on the Boolean cube is a function from $\{0, 1\}^n$ to \mathbb{R} where $n \in \mathbb{N}$. The analysis of Boolean functions is an intensively investigated area of the theoretical computer science. Sometimes the domain is vary; instead of $\{0, 1\}^n$ one can use $\{-1, 1\}^n$, or \mathbb{F}_2^n , an n -dimensional vector space. Some simple example is the *majority*, *parity*, *modular* functions e.t.c

The main tool to represent Boolean functions is the Fourier-Walsh representation. See details in [O'D]

Recently there is a strong interplay between computation complexity and combinatorial number theory. Here just a few example of work of A. Samorodnitsky, L. Trevisan [ST06], an excellent book on Communication Complexity, by A. Rao and A. Yehudayoff [RY20] and recent work of the author [HE20],

2 Boolean functions defined on pseudo-recursive sequences

In the present work we use a structure from additive combinatorics to define a Boolean function and its classical behaviours in computer sciences. These number theoretical structures pop up in additive representation theory. The original question of Erdős and Graham [EG80] remained open (see also [H89]).

Definition 1 *Let $x_1 = a \in \mathbb{N}$, $\{m_i\}_{i=1}^\infty$ be an infinite, $\{b_1, b_2, \dots, b_s\}$ be a finite set of integers. We say that a sequence of integers $X = \{x_i\}_{i=1}^\infty$ is said to be pseudo-recursive sequence if the identity $x_{n+1} = m_{n+1}x_n + b_{j_{n+1}}$ holds, where $b_{j_{n+1}} \in \{b_1, b_2, \dots, b_s\}$ for $n \geq 0$.*

For example the sequence $A_\alpha^p = \{[p^n \alpha] : n \in \mathbb{N}\}$, $p \in \mathbb{N}$; $p \geq 2$; $\alpha \in \mathbb{R}^+$ fulfills this condition. This type of sequences has a long list in the literature (see only e.g. [EG80], [DK68], [BP21], [S21]).

Generally one can define a hypergraph on $[n]$ where the elements of an edge correspond to the variables. We are going to concentrate on some hypergraphs in which there are many pairs of edges with "large" intersections, and in an opposite situation, when the "total intersection" is small.

2.1 Highly intersecting hypergraphs

In this part we consider hypergraphs in which there are many pairs of edges with "large" intersections. More precisely our graph is a k -uniform cycle hypergraph, with $k - 1$ many common elements in the connected edges. In this subsection our sequence will be $\{a_i\}_{i=1}^r = \{\lfloor 2^i \alpha \rfloor\}_{i=1}^r$ $\alpha \geq 1$.

Our first function will be the following: for $x = (x_1, x_2, \dots, x_n) \in \{0, 1\}^n$ let

$$F_\alpha(x_1, x_2, \dots, x_n) := \sum_{j=1}^n a_j \langle x, v^{(j)} \rangle \pmod{2}. \quad (1)$$

where $v^{(j)} = (v_1, v_2, \dots, v_n) \in \{0, 1\}^n$, $v_j = v_{j+1} = \dots, v_{j+k-1} = 1$ and other bits are zero. Throughout the paper we mean $v_s = v_t$ when $s \equiv t \pmod{n}$ and the addition is in \mathbb{Z}_2 .

Note that sometimes this function can also be considered as read-once function, introducing a new variable $y_j = \bigoplus_{i=j}^{j+k-1} x_i$; namely when $\gcd(k, n) = 1$. Recall that a Boolean function is said to be *read-once function*, if all variable appears only one times. So we assume that $\gcd(k, n) > 1$.

We can define a related threshold function too. It will be $T_\alpha(x) := \text{sign}\{G_\alpha\}$, ($\text{sign}\{x\} = 1$ if $x > 0$ and $\text{sign}\{x\} = -1$ otherwise), where $G_\alpha : \{-1, 1\}^n \rightarrow \mathbb{R}$ defined by

$$G_\alpha(x_1, x_2, \dots, x_n) := a_1 x_1 x_2 \cdots x_k + a_2 x_2 x_3 \cdots x_{k+1} + \dots + a_{n-k+1} x_{n-k+1} x_{n-k+2} \cdots x_n + \\ + a_{n-k+2} x_{n-k+2} x_{n-k+3} \cdots x_{n+1} + \dots + a_n x_n x_1 \dots x_{k-1}.$$

2.2 Functions associated to ε -thin sets

Another type of representation of Boolean function when $f : \mathbb{F}_2^n \rightarrow \mathbb{R}$ is the following: $f(x) = \sum_{S \subseteq [n]} \widehat{f}(S) \prod_{i \in S} (-1)^{x_i}$.

Our next function will be the opposite of the previous function; we will assume the total amount of intersections is "small".

The aim of this section is to give an estimation to the cardinality of the image set \overline{H} of $H(x)$.

Definition 2 *The system of sets $\mathcal{S} = \{S_1, S_2, \dots, S_r\}; S_i \in \{0, 1\}^n$, is said to be ε -thin system, if $\sum_{1 \leq i < j \leq r} |S_i \cap S_j| < \varepsilon r$.*

3 Results

Definition 3 *A function is said to be a junta, if it depends only on a fix number of variables.*

Firstly we state that the threshold function is a junta:

Theorem 4 *Let $T_\alpha := \text{sign}\{G_\alpha\}$. The function T_α is a junta.*

Definition 5 For a function g the influence of coordinate i on g is defined as $\text{Inf}_i(g) = \Pr_{x \in \{0,1\}^n} [g(x) \neq g(x \oplus e_i)]$, where x is uniformly distributed over $\{0,1\}^n$, and $g(x \oplus e_i)$ means that we change the i^{th} coordinate to 1 if $x_i = 0$, and to 0 if $x_i = 1$ respectively. The total influence of g is defined to be $\text{Inf}(g) := \sum_i \text{Inf}_i(g)$.

It is an easy exercise that $\text{Inf}_i(g) = \sum_{S \in [n]} \widehat{g}^2(S)$ and $\text{Inf}(g) = \sum_{S \in [n]} |S| \widehat{g}^2(S)$.

Theorem 6 Let $\eta_i \in \{0,1\}$. Then $\text{Inf}_i(F_\alpha) = \eta_i$ if and only if $\sum_{j=i-k+1}^i \alpha_j \equiv \eta_i \pmod{2}$.

As a simple consequence of this we can describe those $S \in [n]$ for which $\widehat{F}_\alpha(S) = 0$. Indeed let $k \leq i \leq n$. Then for every $i \in S$, $\widehat{F}_\alpha(S) = 0$ holds, if and only if $\sum_{j=i-k+1}^i \alpha_j \equiv 0 \pmod{2}$.

In the next result we estimate the chance that F_α has influence at least r .

Theorem 7 Drawn α uniformly at random from $[0,1]$. Then $\text{Inf}(F_\alpha) > r$ holds with probability at least

$$1 - \frac{1}{2^{n-(k-1)r-r \log n}}.$$

A circuit is said to be $AC^0[d]$ -circuit if it consists of AND, OR and NOT gates, fan-in to the gates is unbounded with inputs x_1, x_2, \dots, x_n . The the number of the gates (size of the circuit) is bounded by a polynomial in n , and its depth is at most d .

Corollary 8 Drawn uniformly at random α from $[0,1]$. Then with probability at least

$$1 - \frac{1}{2^{n-k(\Omega((\log n)^d + \log n))}},$$

F_α does not belong to the class $AC^0[d]$.

If $f : \{0,1\}^n \mapsto \{-1,1\}$ is a Boolean function, then its Shannon entropy we mean

$$\mathbb{H}(f) := \sum_S \widehat{f}^2(S) \log \frac{1}{\widehat{f}^2(S)}.$$

The Fourier-Entropy-Influence Conjecture (Briefly FEI conjecture) has a long list in the literature. It states that $\mathbb{H}(f) \ll I(f)$ (recall $I(f)$ is the total influence).

It has proved many type of Boolean functions, e.g. for read-once functions, linear threshold function with high probability e.t.c. Now we will see a corollary of the previous theorem, showing that with high probability for F_α the FEI conjecture holds.

Corollary 9 Drawn uniformly at random α from $[0,1]$. Then with probability at least $1 - \frac{1}{2^{n-O(k^2-k \log n)}}$ the FEI conjecture is true for F_α .

For functions associated to ε -thin sets we are going to define the following function:

Let

$$H(x_1, x_2, \dots, x_n) := \sum_{i=1}^r c_i \prod_{j \in S_i} (-1)^{x_j},$$

where $\{c_i\}_{i=1}^r = \{\lfloor 3^i \alpha \rfloor\}_{i=1}^r$ $\alpha \geq 1$.

First note that all sums in the form $\sum_{i=1}^r \varepsilon_i c_i$; $\varepsilon_i \in \{-1, 1\}$ are pairwise distinct. It immediately implies that $|\overline{H}| \leq 2^r$. The following theorem shows that the lower bound is close the upper bound:

Theorem 10 *Let $H(x_1, x_2, \dots, x_n) := \sum_{i=1}^r c_i \prod_{j \in S_i} (-1)^{x_j}$ where $\{S_i\}_{i=1}^r$ is an ε -thin system. Then*

$$2^{(1-\varepsilon)r} \leq |\overline{H}| \leq 2^r.$$

Acknowledgement. This work is supported by grant K-129335 and the European Union, co-financed by the European Social Fund (EFOP-3.6.3-VEKOP-16-2017-00002).

References

- [BP21] B. Bakos and M. Pálffy Some Results on an Encryption Method Using Subset-Sums of Pseudo-Recursive Sequences, *Discrete Mathematics Letters*, 5. (2021) Pages: 63–67 DOI: 10.47443/dml.2021.0019
- [EG80] P. Erdős, R. L. Graham: Old and new problems and results in combinatorial number theory: van der Waerden’s theorem and related topics, *Enseign. Math.* (2) 25 (1979) no. 3–4, 325–344 (MR81f:10005)
- [H89] N. Hegyvári, Some remarks on a problem of Erdős and Graham. *Acta Math. Hungar.* 53 (1989), no. 1-2, 149–154.
- [H20] N. Hegyvári, On uncertainty inequalities related to subcube partitions and additive energy, arXiv:2009.10127 [cs.DM]
- [DK68] D.E. Knuth, *The Art of Computer Programming, Volume 2*, Addison-Wesley (1968)
- [D] Ryan O’Donnell, *Analysis of Boolean Functions*, Cambridge University Press, 2014
- [RY20] A. Rao, A. Yehudayoff, *Communication Complexity*, <https://homes.cs.washington.edu/~anuprao/pubs/book.pdf>
- [S21] I.D.Shkredov, On multiplicative Chung-Diaconis-Graham process, arXiv:2106.09615v1 [math.CO]
- [ST06] A. Samorodnitsky, L. Trevisan, Gowers Uniformity, Influence of Variables, and PCPs, STOC’06., arXiv:math/0510264v1 [math.CO]

The computational complexity of recognizing some number theoretic properties

Richárd Palincza

Department of Computer Science and Information Theory, Budapest University of
Technology and Economics,
MTA-BME Lendület Arithmetic Combinatorics Research Group
pricsi@cs.bme.hu

1 Introduction

A set of integers is called *primitive* if it does not contain an element dividing another. Similarly a set S of integers is called h -primitive (for a given positive integer h) if there is no distinct elements $a_0, a_1, a_2, \dots, a_h \in S$ with a_0 dividing $a_1 a_2 \dots a_h$. A set B is called a multiplicative basis of order h in $S = \{1, 2, \dots, n\}$ if every member of S can be expressed as the product of at most h (not necessarily distinct) members of B .

2 Primitive sets

The number of primitive subsets of $[n] = \{1, 2, \dots, n\}$ were first studied by Cameron and Erdős [3]. Let $g(n)$ be the number of primitive subsets of $[n]$. They proved that for sufficiently large n , $1.55967^n \leq g(n) \leq 1.6^n$ and conjectured that the limit of $g(n)^{1/n}$ exists. In 2018, Angelo [1] verified this conjecture. However he was unable to provide a method to find better estimate of the limit. With Liu and Pach we [6] proved that the number of primitive subsets of $[n]$ is $(\alpha + o(1))^n$ and we gave an algorithm which can approximate the constant $\alpha \approx 1.57$.

Another related problem is the number of maximum size primitive subsets of $[2n]$. That problem was raised by Bishnoi in his blog post “On a famous pigeonhole problem” [2]. Note that a maximum primitive subset of $[2n]$ is of size n . Indeed, group elements of $[2n]$ into n classes according to their largest odd divisor, then a primitive set can have at most one element from each class. On the other hand $\{n + 1, \dots, 2n\}$ is primitive. Let us denote by $f(n)$ the number of n -element primitive subsets of $[2n]$. Vijay [10] proved that for sufficiently large n , $1.303^n \leq f(n) \leq 1.408^n$, however it was not clear whether the limit of $f(n)^{1/n}$ exists. In [6] we answered this question, showing that this is indeed the case, i.e. $f(n)^{1/n}$ converges to some β , which is roughly 1.318, and gave an algorithm to approximate β to arbitrary precision. For practical purposes (limited by computing power and running time) we calculated that $1.3183 \leq \beta \leq 1.31843$, and $1.571068 \leq \alpha \leq 1.574445$. Later that year McNew improved our numerical lower bounds slightly by different methods in [7], but their method did not to improve on our upper bounds.

Here we present the main ideas (related to computer science) concerning the calculation of α and β . An easy upper bound of the number of choices for maximum size primitive subsets is the following. Partition the numbers in $[2n]$ into chains according to their largest odd divisor. Each chain is in the form $\{t, 2t, 4t, \dots, 2^m t\}$, for an odd t . A trivial upper

bound for the number of choices is choosing from each chain independently. The cornerstone idea of [6] is the following: if we take the chains $\{t, 2t, 4t, 8t, \dots\}$, $\{3t, 6t, 12t, \dots\}$ and $\{9t, 18t, 36t, \dots\}$, we cannot choose really independently if we want to get a primitive subset. For example by choosing $6t$, we rule out the possibility of choosing small elements from $\{t, 2t, 4t, 8t, \dots\}$, e.g. we cannot choose t and $2t$, since they divide $6t$. We also rule out choosing big elements from $\{9t, 18t, 36t, \dots\}$, since they are divisible by $6t$. So instead of choosing independently we should consider these set of chains as a two dimensional (divisibility) lattice, and the task is to choose elements from each row such that there are no two elements dividing another. In two dimensions the number of such choices can be performed effectively by dynamic programming: each row's number of choices depends only the choice of the previous row. However, considering only two-dimensional lattices does not yield to the convergence of $f(n)$, so there remains a gap in upper and lower bounds. The solution for this is generalising this idea further and grouping together the lattice starting with t with the lattice starting with $5t, 25t$, etc. This yields multi-dimensional divisibility lattices. However, these lattices could yield better results given calculating number of choices of large enough instances, we have not found a way to efficiently calculate this in more than two dimensions. Instead we found out a method to combine the results obtained by calculating large two-dimensional lattices effectively and calculating smaller multi-dimensional lattices in a (near) brute force manner.

3 h -primitive sets

The property of h -primitiveness was first studied by Erdős [4] back in 1938 (there called Property \mathcal{P}_h), who studied the maximum size of a 2-primitive subset of $[n]$. We studied the enumeration problem of h -primitive sets. For $h = 1$ the problem is exactly the primitive case mentioned in the previous section. The case $h > 1$ requires different methods. With Pach we proved in [8] that the number of 2-primitive subsets of $\{1, \dots, n\}$ is $T(n) \cdot e^{\Theta(n^{2/3}/\log n)}$ for a certain function $T(n) \approx (3.517\dots)^{\pi(n)}$. Finally, for $h > 2$ the number of h -primitive subsets is $T(n) \cdot e^{\sqrt{n}(1+o(1))}$. Note that the bounds are tight up to a constant factor in the lower order term in the exponent.

The upper bound (in both of the cases $h = 2$ and $h > 2$ is obtained with the help of multiplicative bases of order h . The main idea is that for any fixed multiplicative basis B (of order h) and any A h -primitive set, there is an injective mapping $\varphi : A \rightarrow B$, such that for any $\varphi(a) = b$ we have $b \mid a$. We could determine which elements got mapped to primes and which elements got mapped to the *small* set of non prime elements of B . By carefully counting the number of possible mappings we get the upper bound for h -primitive sets. For the connection of h -primitive sets and multiplicative bases of order h see [9].

4 Complexity results

In this talk we will discuss some algorithmic complexity questions related to h -primitive sets and multiplicative bases of order h . For a constant h , deciding whether a set is h -primitive or whether it is a multiplicative basis (of order h) in a given set S can be performed in polynomial time by checking all subsets in a brute-force manner. However, when h is also part of the input, we show that deciding h -primitive-ness becomes co-NP-

complete and deciding whether a set B is a multiplicative basis of order h in an arbitrary set S becomes NP-complete.

4.1 Complexity of h -primitiveness

A natural question is to decide whether a given set for a given h is h -primitive or not, preferably in polynomial time. If we consider h as a constant, it can clearly be performed in $O(h \cdot n^{h+1})$ arithmetic operations, where n is the size of the given set by enumerating all possible products of size h .

In practical applications where h is large, this algorithm becomes infeasible. If we consider h as part of the input (i.e. we simultaneously want to solve it for any h), this algorithm is exponential in the size of the input. A natural question is whether there exists a better algorithm which is polynomial even with unrestricted h .

The answer to the previous question turns out to be most likely negative: the problem of deciding h -primitive-ness is co-NP-complete, so it can not be performed in polynomial time, unless $P=NP$, and furthermore there isn't even an efficient witness for h -primitive-ness, unless $NP=co-NP$.

Our problem is clearly in co-NP (the witness is a suitable set of a_0, a_1, \dots, a_h , such that $a_0 \mid a_1 a_2 \cdots a_h$). To show that our problem is co-NP-hard, we have to reduce a known NP-complete problem to the *complement* of h -primitive-ness. A suitable problem is the *minimum vertex cover* problem, one of Karp's original 21 NP-complete problems [5]. The minimum vertex cover problem is the following: given a (simple) graph and a positive integer m : the question is whether there is a set S of vertices such that $|S| = m$ and each edge has an endpoint in S (so each edge is "covered" by the vertex set). The main idea of the reduction is to assign numbers to the vertices and take another large number (the "target") such that the "target" number divides the product of h other members if and only if the other members form a vertex cover. Therefore, our set is *not* h -primitive if and only if there is a vertex cover of size h .

4.2 Complexity of multiplicative bases of order h

For a constant h , checking whether a set is a multiplicative basis of order h can be performed similarly like in the h -primitive case: checking all products of size at most h , and checking whether we got all desired elements as products. Like the h -primitive case this is also exponential if we consider h as part of the input.

Another interesting case is when we are interested in multiplicative bases in the ground set $[n]$ (and h is part of the input). It turns out, that even this case is solvable in polynomial time. The algorithm is the following:

1. Take all members of B and mark them as "solved by 1 element"
2. In an iteration take all previously solved elements and multiply them by each member of B . Discard the numbers larger than n , mark the other previously unsolved elements as "solved by size 2 products"
3. Repeat this iteratively: take all newly solved elements (by size i products) and multiply them by each member of B and mark the gotten numbers as "solved by size $i + 1$ ".

4. Iterate this h times, finally labelling elements by “solved by size h products”.
5. Check whether all elements of $[n]$ are solved by a product. If yes, then B is a multiplicative basis of order h otherwise it is not.

This algorithm has a running time of $O(nh|B|)$. At first glance it is not clear whether this is polynomial since n and h can be exponential in the *size* of the input. However, since all multiplicative bases for $[n]$ should contain the $\approx n/\log n$ primes up to n , therefore n is clearly polynomial in the input size. The case of h is similar: if h is larger than $\log_2 n$, a set is an order h multiplicative basis, then it is also a multiplicative basis of order $\log_2 n$, so it is enough to check until that point. So effectively (with small modifications) this is a polynomial algorithm.

However when h is part of the input and we consider any given set (not only in the form of $\{1, 2, \dots, n\}$), the problem becomes NP-complete, even if the given set is a singleton (i.e. asking whether a single element can be expressed as the product of at most h others). The idea of the proof is to reduce the X3C (exact cover by size 3 sets) problem to it. That problem is also part of the Karp’s original 21 NP-complete problems.

References

- [1] **R. Angelo**, A Cameron and Erdős conjecture on counting primitive sets, *Integers 18 (2018)*, A25, 4pp.
- [2] **A. Bishnoi**, <https://anuragbishnoi.wordpress.com/2017/11/02/on-a-famous-pigeonhole-problem>
- [3] **P. J. Cameron and P. Erdős**, On the number of sets of integers with various properties, *Number Theory (Banff, AB, 1988)*, 61–79., de Gruyter, Berlin (1990).
- [4] **P. Erdős**, On sequences of integers no one of which divides the product of two others and on some related problems, *Tomsk. Gos. Univ. Uchen. Zap.*, 2, (1938), 74–82.
- [5] **R. Karp**, Reducibility among combinatorial problems *Complexity of Computer Computations*, Plenum Press (1972) 85–103.
- [6] **H. Liu, P. P. Pach, R. Palincza**, The number of maximum primitive sets of integers, *Combinatorics, Probability and Computing*, 1-15. (2021) doi:10.1017/S0963548321000018
- [7] **N. McNew**, Counting primitive subsets and other statistics of the divisor graph of $\{1, 2, \dots, n\}$, *European Journal of Combinatorics*, 92 (2021) 103237
- [8] **P. P. Pach, R. Palincza**, The counting version of a problem of Erdős, *European Journal of Combinatorics*, 90 (2020) 103187
- [9] **P. P. Pach, Cs. Sándor**, Multiplicative bases and an Erdős problem, *Combinatorica*, 38 (5), (2018), 1175–1203.
- [10] **S. Vijay**, On large primitive subsets of $\{1, 2, \dots, 2n\}$, arXiv:1804.01740

Section:

Coding theory and applications in cryptology

Organizer: György Kiss

Invited talk:

Marcella Takáts, Máté Gyarmati, Péter Ligeti and Péter Sziklai: Secret sharing, coding theory and finite geometry

Contributions:

- Sabira El Khalfaoui and Gábor P. Nagy: Selecting secure parameters of Hermitian subfield subcodes for post-quantum schemes
- Tamás Héger, Zoltán Lóránt Nagy: Short minimal codes and covering codes through geometric and probabilistic methods
- Rebeka Kiss and Gábor P. Nagy: Correlation-immune Boolean functions and parameters of orthogonal arrays
- Sára Pituk: MCF codes and multiple saturating sets



Secret sharing, coding theory and finite geometry

Marcella Takáts*, Máté Gyarmati, Péter Ligeti and Péter Sziklai

ELKH-ELTE Geometric and Algebraic Combinatorics Research Group

marcella.takats@ttk.elte.hu

Secret sharing refers to methods for distributing some secret information amongst a finite set of participants holding a partial information of the secret called share. The goal is to distribute these shares in such a way that only predefined coalitions of users are able to compute the secret.

Several secret sharing constructions are based on geometric objects. In this talk we investigate multilevel schemes, where the participants are partitioned into groups of the same role. Especially, we propose finite geometric constructions for compartmented and conjunctive hierarchical secret sharing schemes.

Within this talk we consider secret sharing schemes from an algorithmic point of view. Assume that some secret information s is distributed amongst a group of participants \mathcal{P} by a special additional entity called dealer. The dealer participates in this distribution step only. The secret s can be reconstructed from the respective share only when a sufficient number of shares are combined together. The collection of possible "reconstructers" is described by the so-called access structure \mathcal{A} , i.e. a monotone increasing set of subsets of the participants. The talk is based on [4] and [5].

In this talk we use the following useful linear algebraic method introduced by Blakley and Kabatianskii [1] and van Dijk [2]. Let us assume that the dealer and the participants are assigned vectors $d, v_i \in \mathbb{F}_q^k$ for $i \in \mathcal{P}$. The proposed constructions are based on the following result:

Theorem 1 (Blakley and Kabatianskii [1]) *A linear secret sharing generated by $G = (d, v_1, \dots, v_{|\mathcal{P}|})$ represents an ideal perfect secret sharing scheme realizing \mathcal{A} if and only if the following conditions hold:*

1. $\forall X \in \mathcal{A}$ the vector d is a linear combination of the vectors $v_x, x \in X$;
2. $\forall Y \notin \mathcal{A}$ the vector d is disjoint from the subspace generated by vectors $v_y, y \in Y$.

Multilevel secret sharing is one straightforward generalization of the widely used t -threshold schemes, where, apart from some threshold value(s), the set of participants is partitioned into smaller subsets (called groups or levels) such that the users within any given level are equivalent from the secret sharing point of view. We are focusing on two special cases, namely on compartmented access structures with upper bounds and on hierarchical threshold access structures as a generalization of results [4]. Further general multilevel constructions based on bivariate interpolation techniques are introduced by Tassa and Dyn [6].

In *compartmented access structures with upper bounds* the goal is to avoid a given percentage of members from all (disjoint) groups in qualified subsets. More precisely, let

$\mathcal{P} = \bigcup_{i=1}^m \mathcal{G}_i$ and let $t \in \mathbb{N}, t_i \in \mathbb{N}, i = 1, \dots, m$ be thresholds with $t \leq \sum_{i=1}^m t_i$. Then the access structure is the following:

$$\mathcal{A} = \{A \subseteq \mathcal{P} : \exists B \subseteq A \text{ such that } |B \cap \mathcal{G}_i| \leq t_i, \forall 1 \leq i \leq m \text{ and } |B| = t\}.$$

We propose geometric constructions for the special case of $t_1 = \dots = t_m = t - 1$ and show the limits of this method as well.

In *hierarchical* threshold access structures with m disjoint levels, let $\mathcal{P} = \bigcup_{i=1}^m \mathcal{L}_i$ and let $t_1 < t_2 < \dots < t_m$ be a sequence of thresholds.

In *conjunctive* (t_1, \dots, t_m) -*hierarchical schemes* the access structure is the following:

$$\mathcal{A} = \left\{ A \subseteq \mathcal{P} : |A \cap \left(\bigcup_{j=1}^i \mathcal{L}_j \right)| \geq t_i, \text{ for all } 1 \leq i \leq m \right\}.$$

Only some sporadic constructions are known for conjunctive hierarchical schemes ([6]).

We suggest ideal constructions for special cases of hierarchical access structures, in particular a 2-level conjunctive $(1, n + 1)$ -hierarchical scheme and 3-level conjunctive $(1, 2, n + 1)$ scheme using finite geometry arguments. We propose ideas for generalization of these constructions for any number of levels.

Let $\text{PG}(n, q)$ denote the projective space of dimension n over the finite field \mathbb{F}_q . Π_r will be the shorthand for a projective subspace of dimension r . A *pencil* in Π_r is the set of the $(q + 1)$ Π_{r-1} -s (in the fixed Π_r), each containing a common fixed Π_{r-2} . A set of t points ($1 \leq t \leq n + 1$) in $\text{PG}(n, q)$ is *independent* if no Π_{t-2} contain them. A set of k points in $\text{PG}(n, q)$ is a *k-arc* if any subset of size $n + 1$ is independent. Note that if $n = 1$ it means that in $\text{PG}(1, q)$ any set of points is an arc. The following configuration defined in [3] is the key to our constructions:

Definition 2 Let Ψ_0, \dots, Ψ_q be a pencil through some Π_{n-2} in $\text{PG}(n, q)$. A *pencil arc* (*k-parc*) \mathcal{K} is a set of k points, in $\text{PG}(n, q)$ satisfying the following conditions:

1. Each $\mathcal{K} \cap \Psi_i$ is a k_i -arc in Π_{n-1} for $0 \leq i \leq q$, where $k_i = |\mathcal{K} \cap \Psi_i|$;
2. $\mathcal{K} \cap \Psi_i \cap \Psi_j = \emptyset$ for $0 \leq i \neq j \leq q$;
3. Any $n + 1$ points of \mathcal{K} not contained in any single Ψ_i are independent.

Note (Fuji-Hara, Miao): if there is a k -parc in $\text{PG}(t - 1, q)$ as above, with $k = k_0 + k_1 + \dots + k_m$ points, $k_i \geq 1$ for $0 \leq i \leq m$ and $k_0 = \min\{k_i\}$, then there exists an ideal secret sharing scheme realizing compartmented access structure with upper bounds $t_1 = \dots = t_m = t - 1$ on $|\mathcal{P}| = k - k_0$ participants.

In [3] it was proved that in $\text{PG}(2, q)$, a k -parc is of size at most $k \leq 2q$. We extend this result to higher dimensions.

Theorem 3 Let \mathcal{K} be a k -parc in $\text{PG}(n, q)$. Then

- (i) if $n = 2$ then $k \leq 2q$, with equality if and only if \mathcal{K} is the point set of two lines minus their intersection point;

(ii) if $n \geq 3$ then $k \leq M(n-1, q) + 1$, where $M(n, q)$ is the largest size of an arc in $\text{PG}(n, q)$.

Note that we have examples of size $k = M(n-1, q) + 1$.

In their paper [3], Fuji-Hara and Miao gave a construction based on Baer subplanes for 2-dimensional pencil arcs. We extend their constructions in different ways. First, we construct pencil arcs from planar arcs.

Identify $\text{AG}(2, q^h) \sim X \times Y$, where $X \sim \mathbb{F}_q^h$ and $Y \sim \mathbb{F}_q^h$ are the horizontal and the vertical axes. Let's call here the translates of the first factor (horizontal axis) the horizontal lines $\ell_0, \dots, \ell_{q^h-1}$, which, together with ℓ_∞ , form the pencil with center P .

Let L_1 be a $(h-1)$ -dim q -subspace of the horizontal axis, i.e. $X = L_0 \times L_1$ for some 1-dim q -vectorspace $L_0 \subset X$, wlog $L_0 = \mathbb{F}_q$. Let L_2 be a 1-dim q -subspace of the vertical axis Y , again wlog $L_2 = \mathbb{F}_q$. Finally, suppose wlog $\ell_0, \dots, \ell_{q-1}$ are the pencil lines intersecting L_2 .

Let $A_0 = L_1 \times L_2$. Any horizontal translate of it is either disjoint from A_0 or identical with it, hence they form a partition $\cup_{\lambda \in \mathbb{F}_q} (A_0 + \lambda) = \ell_0 \cup \dots \cup \ell_{q-1}$. Note that here, for any point $Q \in \ell_0 \cup \dots \cup \ell_{q-1}$, it has coordinates $Q = (a + \lambda, y)$, where $a \in L_1, \lambda \in L_0$ and $y \in L_2$.

Consider the affine plane $\text{AG}(2, q) \sim L_0 \times L_2$ and an arc S in it. Define $K := \{(a + \lambda, y) : a \in L_1, (\lambda, y) \in S\}$.

Observe that K consists of $|S|$ "line segments", each contained in one of the pencil lines ℓ_i and of size $|L_1| = q^{h-1}$.

K is a pencil arc (of size $|S|q^{h-1}$).

There exist arcs of size $q + 1$ in $\text{AG}(2, q)$ for q odd and arcs of size $q + 2$ in $\text{AG}(2, q)$ for q even \implies (many) k -parcs with $k = q^h + q^{h-1}$ in planes of odd order q^h ; and k -parcs with $k = q^h + 2q^{h-1}$ in planes of even order q^h .

We also gave a pencil arc based on caps.

Definition 4 Let Ψ be a hyperplane of $\text{PG}(n, q)$, \mathcal{K}_1 be a set of k_1 points in $\text{PG}(n, q) \setminus \Psi$, and \mathcal{K}_2 be a set of k_2 points in Ψ . A hierarchical arc in $\text{PG}(n, q)$ is a set $\mathcal{K} = \mathcal{K}_1 \cup \mathcal{K}_2$ of $k_1 + k_2$ points in $\text{PG}(n, q)$, also called a (k_1, k_2) -harc, satisfying the following conditions:

- (1) \mathcal{K}_1 is a k_1 -arc in $\text{PG}(n, q)$;
- (2) \mathcal{K}_2 is a k_2 -arc in $\text{PG}(n-1, q)$;
- (3) Any $n + 1$ points of \mathcal{K} not contained in the hyperplane Ψ are independent.

Fuji-Hara and Miao [3] showed that if there is a (k_1, k_2) -harc in $\text{PG}(t-1, q)$ with $k_1 \geq 2$ and $k_2 \geq 0$ then there exists an ideal conjunctive $(1, t)$ -hierarchical scheme with $|\mathcal{P}| = k_1 + k_2 - 1$. The authors also proved that in $\text{PG}(2, q)$ for a (k_1, k_2) -harc its size is at most $k_1 + k_2 \leq q + 2$. The following theorem extends this result to higher dimensions. An affine pointset $S \subset \text{AG}(2, q)$ is called a *hyperfocused arc* if it is an arc and its secants determine $|S| - 1$ directions (which is the least possible value).

Theorem 5 Let \mathcal{K} be a (k_1, k_2) -harc in $\text{PG}(n, q)$, $|\mathcal{K}| = k_1 + k_2 = k$. Then

- (i) if $n = 2$ then $k \leq q + 2$, with equality if and only if \mathcal{K}_1 is a hyperfocused arc of the affine plane and \mathcal{K}_2 is the set of non-determined directions;

(ii) if $n \geq 3$ then $k \leq M(n-1, q) + 1$, where $M(n, q)$ is the largest size of an arc in $\text{PG}(n, q)$.

A conjunctive hierarchical $(1, 2, n+1)$ -scheme ($n \geq 3$): we also gave new constructions for arcs in $\text{PG}(n, q)$. Our construction is a generalization of an ideal conjunctive $(1, n+1)$ -hierarchical scheme based on [3].

A geometric scheme composed of 3 levels $\mathcal{L}_1, \mathcal{L}_2, \mathcal{L}_3$. A valid subset should contain at least $n+1$ elements from $\mathcal{L}_1 \cup \mathcal{L}_2 \cup \mathcal{L}_3$, such that at least 2 elements are from $\mathcal{L}_1 \cup \mathcal{L}_2$ and at least 1 element from \mathcal{L}_1 . In $\text{PG}(n, q) = \text{AG}(n, q) \cup H_\infty$ we will choose our sets as follows. Let

- $|\mathcal{L}_1| = k_1 = c_1 q^{1/n}$ be a subset of an arc (e.g. a so-called normal rational curve) in $\text{AG}(n, q)$;
- $|\mathcal{L}_2| = k_2 = c_2 q^{1/n}$ be a subset of an arc (e.g. a normal rational curve) in H_∞ ;
- $|\mathcal{L}_3| = k_3 = c_3 q^{1/n}$ be a subset of an arc (e.g. a normal rational curve) in H which is a $(n-2)$ -dimensional subspace of H_∞ ;
- furthermore, a set $\mathcal{D} \subset \text{AG}(n, q)$ of size $c_4 q$ is determined, such that the *dealer*, i.e. a point D will be chosen from \mathcal{D} .

Note that, this construction works if $q > cn^n$ yielding an $O(n^3)$ improvement in the size of the underlying field in contrast with the best known general result of Tassa and Dyn [6].

References

- [1] **E.F. Blakley, G.A. Kabatianskii**, Linear algebra approach to secret sharing schemes, *Error Control, Cryptology, and Speech Compression*, **LNCS 829** (1994), 33–40.
- [2] **M. van Dijk**, A linear construction of secret sharing schemes, *Des. Codes Cryptogr.*, **12** (1997), 161–201.
- [3] **R. Fuji-Hara, Y. Miao**, Ideal Secret Sharing Schemes: Yet Another Combinatorial Characterization, Certain Access Structures, and Related Geometric Problems, (2008), URL: <https://infoshako.sk.tsukuba.ac.jp/~fujihara/ftp/sssOct.pdf>
- [4] **P. Ligeti, P. Sziklai, M. Takáts**, Generalized threshold secret sharing and finite geometry, *Des. Codes Cryptogr.*, (2021), DOI 10.1007/s10623-021-00900-9.
- [5] **M. Gyarmati, P. Ligeti, P. Sziklai, M. Takáts**, Conjunctive hierarchical secret sharing by finite geometry, *in preparation*.
- [6] **T. Tassa, N. Dyn**, Multipartite secret sharing by bivariate interpolation, *J. Cryptology* **22** (2009), 227–258.

Selecting secure parameters of Hermitian subfield subcodes for post-quantum schemes

Sabira El Khalfaoui and Gábor P. Nagy

Bolyai Institute, University of Szeged, Aradi vértanúk tere 1, H-6720 Szeged, Hungary
Department of Algebra, Budapest University of Technology and Economics, Egry József
utca 1, H-1111 Budapest, Hungary

sabira@math.u-szeged.hu, nagyg@math.u-szeged.hu

The McEliece cryptosystem is a promising alternative to public-key cryptosystems based on difficult mathematical problems, and it is thought to be secure against post-quantum attacks. The class of subfield subcodes of linear codes yields some good codes, which are of interest because of their applications to public-key cryptography due to McEliece and Niederreiter and to signature schemes based on error-correcting codes. In our previous work [10, 9], we investigated the problem of finding the true dimension of Hermitian subfield subcodes. This inspires us to build McEliece scheme based on these code parameters. Indeed, reducing the key size and improving the security level of the McEliece cryptosystem are among the aims of cryptography today. McEliece cryptosystem is promising for the post-quantum era [2, 14].

This paper's primary goal is to provide a thorough security analysis for the parameter selection process, which involves the computational cost of information set decoding (ISD) algorithm using Hermitian subfield subcodes parameters. Our approach focuses on the optimal parameters that improve the key size for a given security level. Furthermore, due to practical considerations, the key size of several parameter selections is compared to that of the classical McEliece cryptosystem submitted to NIST [3] for the same security level. Besides, we identify the Hermitian subfield subcodes parameters that achieve a Schur square dimension roughly equal to that of random codes. This technique is employed in the so-called distinguisher attack, and that may allow the attacker to determine the Schur square dimension of the code used as a public key.

1 Preliminaries

We refer the reader to [10, 9, 8] for basic concepts on algebraic curves and algebraic geometry (AG) codes such as curves, function fields, valuations, divisors, and Riemann–Roch spaces. In [8], we provide a sketch of the most relevant results on subfield subcodes of linear codes topic. In the sequel of this paper, we use the same notation as in [9].

1.1 Hermitian codes

Hermitian codes are a class of algebraic geometry codes with good properties, they are constructed from Hermitian curves over finite fields. Let \mathcal{H}_q be a Hermitian curve over a finite field \mathbb{F}_{q^2} . \mathcal{H}_q has the form $\mathcal{H}_q : Y^q + Y = X^{q+1}$ in affine coordinates. It is a non-singular curve, and its genus is $g = q(q-1)/2$ by the definition of the genus formula. The points of the projective plane $PG(2, q^2)$ satisfying the homogenous equation

$Y^qZ + YZ^q = X^{q+1}$ are called the rational points of \mathcal{H}_q and denoted by $\mathbb{F}_{q^2}(\mathcal{H}_q)$. \mathcal{H}_q has one infinite point $P_\infty = (0 : 1 : 0)$ and q^3 affine rational points P_1, \dots, P_{q^3} , this make the class of Hermitian curves interesting since they attain the maximal number of rational points for Hasse-Weil bound [11].

By a Hermitian code we mean a functional AG code of the form $C_{\mathcal{L}}(D, G)$. Given all rational points P_1, P_2, \dots, P_{q^3} of \mathcal{H}_q , a divisor D on \mathcal{H}_q is a formal sum $D = P_1 + P_2 + \dots + P_{q^3}$. In this paper, the divisor G takes two forms depending on the type of Hermitian codes. In the 1-point Hermitian code $C_L(D, G)$ case, G has the form $G = sP_\infty$ where s is a positive integer belonging to $\mathbf{S} = \{1, \dots, n + 2g - 1\}$. For the degree 3 place Hermitian code, $G = sP$ where P is a place of degree 3. In the 1-point case, the basis of the Riemann-Roch space $\mathcal{L}(sP_\infty)$ can be given explicitly by [15]:

$$\mathcal{M}(s) := \{x^i y^j \mid 0 \leq i \leq q^2 - 1, 0 \leq j \leq q - 1, qi + (q + 1)j \leq s\}.$$

In the degree 3 case, the Riemann-Roch space

$$\mathcal{L}(sP) = \left\{ \frac{f}{(\ell_1 \ell_2 \ell_3)^u} \mid f \in \mathbb{F}_{q^2}[X, Y], \deg f \leq 3u, v_{P_i}(f) \geq v \right\} \cup \{0\}.$$

can be computed, see [12].

1.2 Hermitian subfield subcodes

Let q be a prime power. We consider the Hermitian curve \mathcal{H}_q over \mathbb{F}_{q^2} , together with the divisor D which is the sum of affine rational points of \mathcal{H}_q . The divisor G is equal either to the rational infinite place P_∞ , or the degree 3 Hermitian place P , respectively. Then, for any integer $s \in \mathbf{S}$ and subfield \mathbb{F}_r of \mathbb{F}_{q^2} , the Hermitian subfield subcodes

$$C_{q,r}^{1\text{-pt}}(s) = C_L(D, sP_\infty)|_{\mathbb{F}_r}, \quad C_{q,r}^{\text{deg-3}}(s) = C_L(D, sP)|_{\mathbb{F}_r}$$

are well defined, and they are \mathbb{F}_r -linear codes of length $n = q^3$, and minimum distance $\delta_\Gamma = n - \deg G$ where δ_Γ is called the Goppa designed minimum distance.

1.3 McEliece cryptosystem

McEliece introduced the first code-based public-key cryptosystem in 1978 where he employed error-correcting codes to generate the public and private key with security relying on two aspects: NP-completeness of decoding linear codes and distinguishing the chosen family of codes. In his original proposal McEliece used binary Goppa codes which are the subfield subcodes of generalized Reed-Solomon codes.

Let C be a linear code of length n , dimension k and minimum distance d , we denote the error capability by $t = \lfloor \frac{d-1}{2} \rfloor$. For the keys generation, we consider the generator matrix usually in its systematic form $G = \begin{bmatrix} I_k \\ G_0 \end{bmatrix}$ of C , a random $k \times k$ invertible matrix S and $n \times n$ permutation matrix P . Thus, the public key is $\mathcal{K}_{pub} = (G', t)$ where $G' = SGP$ and which has size of $k(n - k) \lceil \log_2(q) \rceil$. The secret key is $\mathcal{K}_{sec} = \{G, S, P\}$. Let m be a plaintext of length k , and e a random error vector such that $wt(e) \leq t$.

- **Encryption:** $c = mG' = mSGP + e$.

- **Decryption:** to get back the original message m from c , we simply compute $(mSGP + e)P^{-1} = mSG + eP^{-1}$, then we decode to get mS . Thus $mSS^{-1} = m$.

1.4 Structural and decoding attacks against McEliece cryptosystem

Attacks against code-based cryptography can be divided into two classes: structural or key recovery attacks which aimed at recovering the secret code, and decoding, or message recovery attacks that seek to decrypt the transmitted ciphertext. The most recent and most effective structural attack against AG code-based McEliece cryptosystems is the Schur product distinguisher, which is given in [4, 7, 5].

Definition 1 (Schur product). *Given two elements $a = (a_1, \dots, a_n)$ and $b = (b_1, \dots, b_n)$ in \mathbb{F}_q^n , The Schur product is the component-wise*

$$a * b := (a_1b_1, \dots, a_nb_n)$$

product on \mathbb{F}_q^n . For two linear subspaces $A, B \subseteq \mathbb{F}_q^n$, their Schur product is the linear subspace

$$A * B := \text{Span}_{\mathbb{F}_q} \{a * b \mid a \in A \text{ and } b \in B\}.$$

*If $B = A$, then $A * A$ is denoted as A^{*2} and, we define A^{*t} by induction for any positive integer t .*

Since one of the McEliece cryptosystem's security assumptions is that the chosen family of codes must be indistinguishable. The Schur product operation is a useful tool for distinguishing AG codes from random ones because the evaluation codes do not behave like random ones. More precisely, let $C_L(D, G)$ be an AG code and C a linear random code both in \mathbb{F}_q^n and have the same dimension k , the argument is that $\dim C_L(D, G)^{*2}$ is smaller than $\dim C^{*2} \leq \frac{k(k+1)}{2}$.

The security of the McEliece cryptosystem is also based on the NP-completeness of linear code decoding. In 1962, Prange introduced a generic decoding algorithm called *Information Set Decoding* that can solve computational syndrome decoding (CSD) [13], which consists of correcting t errors that occur in a codeword of an $[n, k]$ linear code and does not require an explicit code structure. We base our security analysis on the time complexity of Prange algorithm:

$$C_{\text{Prange}}(n, k, t) = \frac{\binom{n}{t}}{\binom{n-k}{t}} C_{\text{Gauss}}(n, k), \quad (1)$$

where $C_{\text{Gauss}}(n, k, q)$ is the time complexity of the Gauss-Jordan elimination of a $k \times n$ matrix over \mathbb{F}_q . Many improvements have been made to Prange's algorithms; however, they do not make a significant difference.

2 Our proposal

The family of codes we suggest to use for McEliece scheme is the $\mathbb{F}_{q^2}/\mathbb{F}_q$ subfield subcodes of 1-point and degree 3 place Hermitian codes. The National Institute of Standards and

Technology [3] has recently begun a selection process to standardize quantum-resistant public-key cryptosystems.

We consider classical McEliece cryptosystem variants built on Goppa codes. The parameters for cryptosystems reported in [1] are designed to be comparable to the computational cost required to break *AES-128*(category 1), *AES-192*(Category-3), and *AES-256* (Category 5). The following tables summarize the code parameters of Classical McEliece cryptosystem submitted to *NIST round 2-code-based cryptosystems*, and those of Hermitian subfield subcodes $C = C_{q,r}^\gamma(s)$ (code length n , dimension k , and error-capability t), as well as the computational cost of Prange’s ISD algorithm for each variant, expressed as \log_2 (bit operations) with the public key size.

Classic McEliece	n	k	t	Prange	Key-Size(bit)
Category 1	3488	2720	64	170.82	2088960
Category 3	4608	3360	96	214.70	4193280
Category 5	6688	5024	128	293.55	8359936
	6960	5413	119	294.49	8373911
	8192	6528	128	331.64	10862592

Table 1: Classical McEliece cryptosystem

	Code Type	n	k	t	Prange	Key-Size(bit)
Category 1	$C_{16,4}^{1-pt}$ (2580)	4096	430	757	170.72	3152760
	$C_{16,2}^{1-pt}$ (3474)	4096	1027	310	170.64	3151863
	$C_{11,11}^{1-pt}$ (1174)	1331	927	78	170.66	1295584
	$C_{13,13}^{1-pt}$ (1158)	2197	322	519	170.20	2234140
	$C_{11,11}^{\text{deg}-3}$ (288)	1331	413	233	170.78	1311588
	$C_{13,13}^{\text{deg}-3}$ (389)	2197	325	514	170.05	2251347
Category 3	$C_{16,4}^{1-pt}$ (2729)	4096	620	683	214.25	4310240
	$C_{13,13}^{1-pt}$ (2038)	2197	1733	79	214.85	2975567
	$C_{13,13}^{\text{deg}-3}$ (678)	2197	1717	81	215.52	3049754
Category 5	$C_{13,13}^{1-pt}$ (1860)	2197	1396	168	293.33	4137816
	$C_{13,13}^{\text{deg}-3}$ (614)	2197	1360	177	295.99	4212284

Table 2: McEliece cryptosystem based on Hermitian subfield subcodes

References

- [1] **Marco Baldi, Alessandro Barenghi, Franco Chiaraluce, Gerardo Pelosi, and Paolo Santini**, A Finite Regime Analysis of Information Set Decoding Algorithms, *Algorithms* **12**(10):209, 2009.
- [2] **Frank Arute et al.**, Quantum supremacy using a programmable superconducting processor, *Nature*, **574** (2019), 505–510.

- [3] *Post-Quantum Cryptography*,
<http://csrc.nist.gov/projects/post-quantum-cryptography>. Updated: March 25, 2020
- [4] **Alain Couvreur, Irene Márquez-Corbella, and Ruud Pellikaan**, Cryptanalysis of McEliece cryptosystem based on algebraic geometry codes and their subcodes, *Transactions on Information Theory* **63** (2017), 5404–5418.
- [5] **Alain Couvreur, Ayoub Otmani, and Tillich Jean-Pierre**, Polynomial time attack on wild McEliece over quadratic extensions, *IEEE Transactions on Information Theory* **63** (2016), 404–427.
- [6] **Alain Couvreur, Philippe Gaborit, Valérie Gauthier-Umaña, Ayoub Otmani, and Jean-Pierre Tillich**, Distinguisher-based attacks on public-key cryptosystems using Reed–Solomon codes. *Designs, Codes and Cryptography* **73** (2014), 641–666.
- [7] **Alain Couvreur, Irene Márquez-Corbella, and Ruud Pellikaan**, Cryptanalysis of public-key cryptosystems that use subcodes of algebraic geometry codes, In: Pinto R., Rocha Malonek P., Vettori P. (eds) *Coding Theory and Applications*, CIM Series in Mathematical Sciences, vol. **3**, Springer, Berlin, 2015, 133–140.
- [8] **Sabira El Khalfaoui**, *On the dimension of the subfield subcodes of Hermitian codes*. PhD Thesis, University of Szeged, 2020.
- [9] **P. Nagy, Gábor and Sabira El Khalfaoui**, Estimating The Dimension of the Subfield Subcodes of Hermitian Codes, *Acta Cybernetica* **24** (2019), 625–641.
- [10] **Sabira El Khalfaoui and Gábor P.Nagy**, On the dimension of the subfield subcodes of 1-point Hermitian codes, *Advances in Mathematics of Communications* **15** (2021), 219–226.
- [11] **Alfred J. Menezes, Ian F. Blake, XuHong Gao, Ronald C. Mullin, Scott A. Vanstone, and Tomik Yaghoobian**, *Applications of Finite Fields*, vol. 199. Springer Science & Business Media, 2013.
- [12] **Gábor Korchmáros and Gábor P. Nagy**, Hermitian codes from higher degree places, *J. Pure Appl. Algebra* **217** (2013), 2371–2381.
- [13] **Christiane Peters**, *Information-set decoding for linear codes over \mathbb{F}_q* . In: International Workshop on Post-Quantum Cryptography, Springer, Berlin, 2010, 81–94.
- [14] **P. W. Shor**, Polynomial-time algorithm for prime factorization and discrete logarithms on a quantum computer, *SIAM Journal on Computing* Volume **26I** (1997), 1484–1509.
- [15] **Serguei A. Stepanov**, *Codes on algebraic curves*. Springer Science & Business Media, 2012.



Short minimal codes and covering codes through geometric and probabilistic methods

Tamás Héger, Zoltán Lóránt Nagy

ELKH–ELTE Geometric and Algebraic Combinatorics Research Group, Eötvös Loránd University, Budapest, Hungary

heger.tamas@ttk.elte.hu, nagyzoli@caesar.elte.hu

Throughout this extended abstract, q denotes a prime power and \mathbb{F}_q denotes the Galois field of q elements, while p stands for the characteristics of \mathbb{F}_q . Let \mathbb{F}_q^n be the n -dimensional vector space over \mathbb{F}_q . Denote by $[n, r]_q$ a q -ary linear code of length n and dimension r , which is the set of codewords (code vectors) of a subspace of \mathbb{F}_q^n of dimension r .

Definition 1 *In a linear code, a codeword is minimal if its support does not contain the support of any codeword other than its scalar multiples. A code is minimal if its codewords are all minimal.*

Minimal codewords in linear codes were originally studied in connection with decoding algorithms [10] and have been used by Massey [11] in a secret sharing scheme. For a general overview on recent results in connection with minimal codes we refer to [1]. The general problem is to determine the minimal length of an $[n, k]_q$ minimal code can have, provided that k and q are fixed.

Definition 2 (Minimal length of a minimal code) *Denote by $m(k, q)$ the minimal length of an $[n, k]_q$ minimal code with parameters k and q .*

The following bounds are due to Alfarano et al. [1] and Chabanne et al. [4].

Theorem 3 ([1, 4]) *Let \mathcal{C} be an $[n, k]_q$ minimal code. We have*

$$(k-1)(q+1) \leq m(k, q) \leq \min \left\{ ck^2q, \frac{2k}{\log_q \left(\frac{q^2}{q^2-q+1} \right)} \right\}$$

for some $c \geq 2/9$.

Note that the upper bound of [4] is non-constructive and it roughly says $m(k, q) \lesssim 2kq \ln(q)$, while the quadratic upper bound due to Alfarano et al. [1]. Our contribution is a linear upper bound in both k and q .

Theorem 4

$$m(k, q) \leq \left\lceil \frac{2}{1 + \frac{1}{(q+1)^2 \ln q}} (k-1) \right\rceil (q+1) \quad \text{and} \quad m(k, 2) \leq \frac{2k-1}{\log_2 \left(\frac{4}{3} \right)}.$$

for $q > 2$ and $q = 2$, respectively.

Blocking sets and their generalisations are well-known concepts in finite geometry. As it was noticed recently by Alfarano, Borello and Neri [2] and independently by Tang, Qiu, Liao, and Zhou [12], minimal codes are in one-to-one correspondence with special types of blocking sets of projective spaces, which they called *cutting blocking sets* after the earlier paper of Bonini and Borello [3]. In fact, this concept has been investigated in connection with saturating sets and covering codes a decade earlier by Davydov, Giulietti, Marcugini and Pambianco [5] under the name *strong blocking sets* and in the paper of Fancsali and Sziklai [7] in connection with so-called higgledy-piggledy line arrangements under the name *generator set*. We denote the finite projective geometry of dimension N and order q by $\text{PG}(N, q)$.

Definition 5 (Multifold strong blocking sets, aka cutting blocking sets) *A t -fold strong blocking set of $\text{PG}(N, q)$ is a point set that meets each $(t - 1)$ -dimensional subspace Λ in a set of points which spans the whole subspace Λ [5]. A cutting t -blocking set of $\text{PG}(N, q)$ is a point set that meets each $(N - t)$ -dimensional subspace Λ in a set of points which spans the whole subspace Λ [3]. A cutting blocking set (without prefix) is a cutting 1-blocking set.*

Clearly, cutting t -blocking sets and $(N - t + 1)$ -fold strong blocking sets coincide. It can be shown that a cutting blocking set of $\text{PG}(N, q)$ of size n corresponds to a minimal $[n, N + 1]_q$ code (see [2, 12]). Thus, as short minimal codes are of interest, constructing small cutting blocking sets of $\text{PG}(N, q)$ is highly relevant. One may try construct such a set as the union of lines.

Definition 6 *A set of lines of $\text{PG}(N, q)$ is called higgledy-piggledy, if the union of their point sets is a cutting blocking set of $\text{PG}(N, q)$.*

Fancsali and Sziklai proved the following bounds.

Theorem 7 (Fancsali, Sziklai, [7]) *Let \mathbb{F} be an arbitrary field.*

- i) If $|\mathbb{F}| \geq N + \lfloor N/2 \rfloor$, then every higgledy-piggledy line set of $\text{PG}(N, \mathbb{F})$ contains at least $N + \lfloor N/2 \rfloor$ lines.*
- ii) If $|\mathbb{F}| \geq 2N - 1$, then there exist a higgledy-piggledy line set of $\text{PG}(N, \mathbb{F})$ containing $2N - 1$ lines.*

Note that for $2 \leq N \leq 5$, there are higgledy-piggledy line sets in $\text{PG}(N, q)$ of size $N + \lfloor N/2 \rfloor$, provided that q is large enough. The weakness of Theorem 7 is that it requires q to be large, whereas the typical approach in coding theory is to fix q and let the length of the code vary. The only known construction of higgledy-piggledy line sets that works for general N and q is the so-called tetrahedron: take $N + 1$ points of $\text{PG}(N, q)$ in general position, and then the $\binom{N+1}{2}$ lines joining these points are easily seen to be higgledy-piggledy (see [2, 3, 5]). However, this construction is much larger than the expected minimum.

Thus it is of interest to construct higgledy-piggledy line sets in $\text{PG}(N, q)$ of small size from two points of view. First, they give rise to short minimal codes. Second, to determine whether the lower bound remains valid for small q (and possibly large dimension) as well.

Theorem 4 is derived as follows. Case $q > 2$ is based on a random construction of taking the point set of the union of less than $2k$ lines in a suitable projective space. It turns out that choosing lines uniformly at random results a higgledy-piggledy line set with positive probability. For $q = 2$, the bound is obtained by a randomly selected point set. The expected value of the number of hyperplanes not generated by the set will be less than 1, hence the claim follows.

Finding short minimal codes, that is, small multifold strong blocking sets, has relevance in another code theoretic aspect as well, since multifold strong blocking sets are linked to *covering codes*. From a geometric perspective, these objects correspond to *saturating sets* in projective spaces.

Definition 8 (Saturating sets) *A point set $S \subset \text{PG}(N, q)$ is ρ -saturating if for any point Q of $\text{PG}(N, q) \setminus S$ there exist $\rho+1$ points in S generating a subspace of $\text{PG}(N, q)$ which contains Q , and ρ is the smallest value with this property. Equivalently, the subspaces of dimension ρ which are generated by the $(\rho+1)$ -tuples of S must cover every point of the space. The smallest size of a ρ -saturating set in $\text{PG}(N, q)$ is denoted by $s_q(N, \rho)$.*

For recent upper bounds on ϱ -saturating sets of $\text{PG}(N, q)$ the reader is referred to [6].

Definition 9 (Covering radius, covering code) *The covering radius of an $[n, n-r]_q$ code is the least integer R such that the space \mathbb{F}_q^n is covered by spheres of radius R centered on codewords. If an $[n, n-r]_q$ code has covering radius R , then it is referred to as an $[n, n-r]_q R$ covering code.*

Note that we can apply the following equivalent description. A linear code of co-dimension r has *covering radius* R if every (column) vector of \mathbb{F}_q^r is equal to a linear combination of R columns of a parity check matrix of the code, and R is the smallest value with this property. The covering problem for codes is that of finding codes with small covering radius with respect to their lengths and dimensions. *Covering codes* are those codes which are investigated from the point of view of the above covering problem. Usually the parameters for the covering radius and the co-dimension are fixed and one seeks a good upper bound for the length of the corresponding covering codes.

Definition 10 *The length function $l_q(r, R)$ is the smallest length of a q -ary linear code of co-dimension r and covering radius R .*

There is a one-to-one correspondence between $[n, n-r]_q R$ codes and $(R-1)$ -saturating sets of size n in $\text{PG}(r-1, q)$. This implies $l_q(r, R) = s_q(r-1, R-1)$ [5].

Theorem 11 (Denaux [6], Theorem 6.2.12.) *Suppose that q is a prime power. Then*

$$\frac{\varrho+1}{e} q^{N-\varrho} < s_{q^{e+1}}(N, \varrho) \leq \frac{(\varrho+1)(\varrho+2)}{2} \left(q^{N-\varrho} + \frac{2\varrho}{\varrho+2} \frac{q^{N-\varrho}-1}{q-1} \right).$$

Applying a random construction based on point sets of subspaces in the spirit of higgledy-piggledy line sets, we improved the known upper bounds when q is an R th power and $R \geq \frac{2}{3}r$. The theorem below ensures the existence of a set of higgledy-piggledy $(N-t+1)$ -spaces of size m .

Theorem 12 *There is a strong t -fold blocking set \mathcal{B} in $\text{PG}(N, q)$ consisting of the points of m subspaces of dimension $N - t + 1$ for*

$$m = \lceil (N - t + 2)(t - 1)c_1(q) + c_2(q) \rceil,$$

where the constants $c_1(q)$ and $c_2(q)$ depending on q s.t. $c_1(q) \rightarrow 1$ and $c_2(q) \rightarrow 0$ as q tends to infinity.

Corollary 13 $s_{qe+1}(N, \varrho) \leq \lceil c_1(q)(N - \varrho + 1)\varrho + c_2(q) \rceil \frac{q^{N-\varrho+1}-1}{q-1}$.

Let us also note that the construction of Fancsali and Sziklai [8] yields $s_{qe+1}(N, \varrho) \leq ((N - \varrho + 1)\varrho + 1) \frac{q^{N-\varrho+1}-1}{q-1}$ but requires the condition $q > N + 1$.

References

- [1] **Alfarano, G.N., Borello, M., Neri, A. and Ravagnani, A.**, Three Combinatorial Perspectives on Minimal Codes, *arXiv preprint* (2020), arXiv:2010.16339.
- [2] **Alfarano, G.N., Borello, M. and Neri, A.**, A geometric characterization of minimal codes and their asymptotic performance, *Adv. Math. Commun.*, (2020).
- [3] **Bonini, M. and Borello, M.**, Minimal linear codes arising from blocking sets, *J. Algebraic Combin.*, **53** (2021), 327–341.
- [4] **Chabanne, H., Cohen, G. and Patey, A.**, Towards secure two-party computation from the wire-tap channel, in *Information security and cryptology (ICISC, 2013)*, Lecture Notes in Comput. Sci. **8565**, Springer, Cham, 2014, 34–46.
- [5] **Davydov, A.A., Giulietti, M., Marcugini, S. and Pambianco, F.**, Linear non-binary covering codes and saturating sets in projective spaces, *Adv. Math. Commun.*, **5** (2011), 119–147.
- [6] **Denaux, L.**, Constructing saturating sets in projective spaces using subgeometries, *arXiv preprint* (2020), arXiv:2008.13459.
- [7] **Fancsali, Sz.L. and Sziklai, P.**, Lines in higgledy-piggledy arrangement, *Electron. J. Comb.* **21** (2014), #P2.56.
- [8] **Fancsali, Sz.L. and Sziklai, P.**, Higgledy-piggledy subspaces and uniform subspace designs, *Des. Codes Cryptogr.* **79** (2016), 625–645.
- [9] **Héger, T. and Nagy, Z.L.**, Short minimal codes and covering codes via strong blocking sets in projective spaces, *arXiv preprint* (2021), arXiv:2103.07393.
- [10] **Hwang, T.Y.**, Decoding linear block codes for minimizing word error rate, *IEEE Trans. Inform. Theory*, **25** (1979), 733–737.
- [11] **Massey, J.L.**, Minimal codewords and secret sharing, in: *Proceedings of the 6th joint Swedish-Russian international workshop on information theory* (Mölle, 1993), 276–279.
- [12] **Tang, C., Qiu, Y., Liao, Q. and Zhou, Z.**, Full characterization of minimal linear codes as cutting blocking sets, *IEEE Trans. Inform. Theory*, **67** (2021), 3690–3700.

Correlation-immune Boolean functions and parameters of orthogonal arrays

Rebeka Kiss*, Gábor P. Nagy**

*Bolyai Institute, University of Szeged **Department of Algebra, Budapest University of Technology and Economics

*Kiss.Rebeka@stud.u-szeged.hu **nagyg@math.bme.hu

Boolean functions are often applied in cryptography, since they can be used in stream ciphers as pseudo-random generators. It is natural, that in cryptography one of the most important aim is to reduce the vulnebarility of encryptions. There are several attacks aiming to circumvent the security of a cryptographic system. One of them is the Siegenthaler attack, which uses the existence of correlation between the input and output bits of a Boolean function. Another attack is the side channel attack, which tries to find the weaknesses by analysing physical parameters, like timing information, power consumption, electromagnetic leaks or even sound. We can defend against both of them with correlation immune functions.

Definition 1 (Correlation-immune function) *An $f: \mathbb{F}_2^k \rightarrow \mathbb{F}_2$ Boolean function is said to be **correlation immune of order t** ($1 \leq t \leq k$) with notation **CI(t)**, if for any fixed subset of t variables the probability that, given the value of $f(x)$, the t variables have any fixed set of values is always 2^{-t} , no matter what the choice of the fixed set of t values is.*

An example:

$f: \mathbb{F}_2^4 \rightarrow \mathbb{Z}_2, f(x_1, x_2, x_3, x_4) = x_1 + x_3 + x_4$ is a second degree correlation immune function (CI(2)).

The support of f :

x_1	x_2	x_3	x_4
1	0	0	0
1	1	0	0
0	0	1	0
0	1	1	0
0	0	0	1
0	1	0	1
1	0	1	1
1	1	1	1

Although we can defend against the mentioned attacks, the defense is costly. We would like to know the minimal size of the support of correlation immune functions with k variables and order t in order to reduce the cost. It is difficult to answer in the language of functions, it is easier to examine orthogonal arrays, that are strongly connected to CI-functions. Supports of t -th order correlation immune functions give simple orthogonal arrays with strength t , if their elements are written as rows.

Definition 2 *A binary array with N rows and k columns is said to be an **orthogonal array** if in every subset of the columns with t elements every binary t -tuple appears in exactly $N/2^t$ rows, where t is called the **strength** of this orthogonal array.*

Some orthogonal arrays have a special property. In simple orthogonal arrays there is no repetition among the rows.

The support of the previous function f :

$$\begin{array}{cccc} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{array}$$

We have to check the orthogonal array property with strength 2, which means that every 2-tuple appears exactly twice in every arbitrarily chosen subarray with two columns.

As it was mentioned we would like to know the minimum weight of CI functions. From the aspect of orthogonal arrays the question is that with given parameters k and t what is the minimal value N for which an orthogonal array exists with N rows. The problem now is to tell whether an orthogonal array exists with given parameters. This is difficult already for small parameters. We always look for simple orthogonal arrays because these are the ones that are applied in cryptography.

There is a connection between orthogonal arrays and linear codes, which can help us to give answer in questionable cases.

Theorem 3 *If C is a $(k, N, d)_2$ linear code over the field with two elements with dual distance d^\perp then the codewords of C form the rows of an $OA(N, k, 2, d^\perp - 1)$. Conversely the code corresponding to an $OA(N, k, 2, t)$ is a $(k, N, d)_2$ for some d with dual distance $t + 1$.*

Since linear codes have been studied for a long time, there are a lot of information about them, so in some cases it can be enough to work with them and then give the sought value for orthogonal arrays using the previous lemma.

Some references about correlation immune functions are written by Claude Carlet and his coauthors [2, 3]. They were looking for minimal weights of CI-functions with given parameters, and published a table with these values. There were cases when the answer was unknown. Two of them ($k = 12, t = 6$ and $k = 11, t = 4$) were included also as problems at the NSUCRYPTO International Olympiad in Cryptography [5, 8].

In case $t=2$ there is a connection between orthogonal arrays and Hadamard matrices.

Theorem 4 (Orthogonal arrays and Hadamard matrices) *An $OA(4\lambda, 4\lambda - 1, 2, 2)$ (or equivalently $OA(8\lambda, 4\lambda, 2, 3)$) orthogonal array exists if and only if there exists a Hadamard matrix of order 4λ .*

Already with $t=2$ arises a serious, basic unsolved problem in discrete mathematics.

Conjecture 5 (Hadamard Conjecture) *A Hadamard matrix of order k exists if k is 1, 2 or a multiple of 4.*

An important open problem regarding simple orthogonal arrays is due to Claude Carlet.

Conjecture 6 (Claude Carlet’s Conjecture) *For a fixed strength t the minimal size of the support of k -variable, t -th order correlation-immune functions does not decrease when k grows.*

The statement is trivially true for general orthogonal arrays, since by omitting at most $k - t$ columns we get new orthogonal arrays with less columns, but with the same number of rows and strength. The difficulty lies in that we have to produce orthogonal arrays that do not contain any row multiple times.

We were able to fill out all of the missing entries of the table. Using the solution of the case $k = 11$, $t = 4$ we could give the sought values for further parameters too [7].

One of our results was given by using linear codes, in case $k = 13$, $t = 6$ the minimal number N for which an $OA(N, k, 2, t)$ exists is 1024.

We also used a theorem given by Hedayat, Sloane and Stuffken [6].

Theorem 7 *An $OA(N, k, 2, 2u)$ exists if and only if an $OA(2N, k + 1, 2, 2u + 1)$ exists.*

The properties and theorems were stated for orthogonal arrays in general, our task was to prove that they are also true for simple orthogonal arrays. For solving the problem we used Bulutoglu’s and Margot’s linear programming method [1], but for larger parameters the running time was too long. One of our ideas was to try finding an orthogonal array with given parameters with a special property. We supposed that it had a given automorphism group. With this assumption we were able to decrease the number of conditions and as a result we reduced the running time of the ILP method. We implemented it in Sage [9] and used SCIP [4] to solve the linear programming problems. Our future goal is to give the minimum value in other unsolved cases and also try to solve the Conjecture of Claude Carlet.

References

- [1] **Bulutoglu, D. A. and Margot, F.**, Classification of orthogonal arrays by integer programming, *Journal of Statistical Planning and Inference*, **138:3** (2008), 654–666.
- [2] **Carlet, C. and Chen, X.**, Constructing Low-Weight d th-Order Correlation-Immune Boolean Functions Through the Fourier-Hadamard Transform, *IEEE Transactions on Information Theory*, **64:4** (2018), 2969–2978.
- [3] **Carlet, C. and Guilley, S.**, Correlation-immune Boolean functions for easing counter measures to side-channel attacks, in: Niederreiter, H., Ostafe, A., Panario, D., Winterhof, A. (Eds.) *Algebraic Curves and Finite Fields Cryptography and Other Applications*, Radon Series on Computational and Applied Mathematics **16**, De Gruyter, Berlin, 2014, 41–70.
- [4] **Gamrath, G., Anderson, D., Bestuzheva, K., Chen, W.-K., Eifler, L., Gasse, M., Gemander, P., Gleixner, A., Gottwald, L., Halbig, K., Hendel, G., Honjny, K., Koch, T., Le Bodic, P., Maher, S. J., Matter, F., Miltenberger, M., Mühmer, E., Müller, B., Pfetsch, M. E., Schlösser, F., Serrano, F., Shinano, Y., Tawfik, C., Vigerske, S., Wegscheider, F., Weninger, D. and Witzig, J.** The SCIP Optimization Suite 7.0 (2020), www.optimization-online.org/
- [5] **Gorodilova, A., Agievich, S., Carlet, C., Hou, X.-D., Idrisova, V., Kolomee, N., Kutsenko, A., Mariot, L., Oblaukhov, A., Picek, S., Preneel, B., Rosie, R. and Tokareva, N.**, The Fifth International Students’ Olympiad in cryptography—NSUCRYPTO: Problems and their solutions, *Cryptologia* **44:3** (2020), 223–256.

- [6] **Hedayat, A. S., Sloane, N. J. A. and Stufken, J.**, *Orthogonal Arrays: Theory and Applications*, Springer-Verlag, New York, 1999.
- [7] **Kiss, R., Nagy, G. P.** On the nonexistence of certain orthogonal arrays of strength four, *Prikl. Diskr. Mat.* **52** (2021), 65–68.
- [8] NSUCRYPTO Unsolved problems, www.nsucrypto.nsu.ru/unsolved-problems/
- [9] **The Sage Developers**, SageMath, the Sage Mathematics Software System (Version 9.1) (2020), www.sagemath.org

MCF codes and multiple saturating sets

Sára Pituk

Institute of Mathematics, Eötvös Loránd University

pituksari@gmail.com

1 Introduction

In coding theory, a covering code is a subset $C \subset GF(q)^n$ such that every element of $GF(q)^n$ is within a fixed distance from C , according to the Hamming metric d_H . This fixed value is the covering radius of the code. In some applications (e.g. in list decoding or in the generalised football pool problem) it is useful to consider codes with covering radius R which have the following additional property: for every $x \in GF(q)^n$ such that $d_H(x, C) = R$, the number of codewords in C which have Hamming distance exactly R from x , is at least μ . Such codes are called (R, μ) -MCF codes.

The theory of covering codes is a widely investigated area in coding theory, which has many applications to other fields of mathematics as well. For instance, there is a well-known correspondence between linear covering codes and saturating sets in finite projective spaces. This correspondence can be extended to $(\rho+1, \mu)$ -MCF codes and (ρ, μ) -saturating sets (saturating sets with a certain extra property). This remark lets us apply geometric techniques when dealing with MCF codes.

One way to construct a new linear code from existing ones is to take their direct sum. In this extended abstract, we investigate how this construction affects the parameters of MCF codes. We also examine the density of the resulting MCF code, which is the generalization of the density of covering codes, and measures the optimality of the code.

2 MCF codes

We will denote the finite field of order q by $GF(q)$ and the n -dimensional vector space over $GF(q)$ by $GF(q)^n$. The vectors in $GF(q)^n$ are sometimes also called *words*. The projective space $PG(n-1, q)$ arises from the vector space $GF(q)^n$:

$$PG(n-1, q) = (GF(q)^n \setminus \{0\}) / \sim,$$

where \sim denotes the following equivalence relation: $u \sim v$ if and only if $u = \lambda v$ with some $0 \neq \lambda \in GF(q)$.

We now give some basic definitions from coding theory.

Definition 1 C is a linear $[n, k]_q$ -code if it is a k -dimensional subspace of $GF(q)^n$. The elements of the code are called codewords. The parameters n and k are also referred to as the length and the dimension of the code, respectively. The codimension of the code is $d = n - k$.

Definition 2 Let C be a linear $[n, k]_q$ -code. $M \in GF(q)^{(n-k) \times n}$ is a parity check matrix of C if

$$c \in C \Leftrightarrow c \cdot M^T = 0.$$

To measure the difference between words in $GF(q)^n$ we use the following notion of distance, which is indeed a metric on $GF(q)^n$:

Definition 3 Let x and y be two vectors in $GF(q)^n$. Their Hamming distance $d_H(x, y)$ is the number of their different coordinates.

Definition 4 Let C be a linear $[n, k]_q$ -code and R a positive integer. C is R -covering if for every word $y \in GF(q)^n$ there is a codeword $x \in C$ such that the Hamming distance $d_H(x, y) \leq R$. The covering radius of C is the smallest R such that C is R -covering.

For a word x and an integer R let $\overline{S}(x, R)$ be the closed Hamming ball with radius R and centered in x :

$$\overline{S}(x, R) = \{y \in GF(q)^n : d_H(x, y) \leq R\}.$$

We are now ready to define MCF codes:

Definition 5 A linear $[n, k]_q$ -code C is (R, μ) -MCF (multiple covering of the farthest-off points) if the covering radius of C is R and for every $y \in GF(q)^n$ such that $d_H(y, C) = R$, the value $|\overline{S}(x, R) \cap C|$ is at least μ .

Let

$$V = \{x_1, x_2, \dots, x_{N(C)}\}$$

be the set of vectors in $GF(q)^n$ with distance R from C . We define the μ -density of C as follows:

$$\gamma_\mu(C) = \frac{\sum_{i=1}^{N(C)} |\overline{S}(x_i, R) \cap C|}{\mu N(C)}.$$

Clearly, the μ -density of an (R, μ) -MCF code is at least 1. If equality holds, then every word which has distance R from the code, is covered by exactly μ Hamming balls of radius R around the codewords. A code for which this holds is called APMCF (almost perfect MCF) in the literature. We also introduce the notation AAPMCF (asymptotically almost perfect MCF): this means that the μ -density of the code tends to 1, when q tends to infinity.

MCF codes can be described as geometric objects as follows. Suppose we have a code C , which has length n and codimension d , and let us take a parity check matrix of C . Note that if we multiply a column of the matrix with some $0 \neq \lambda \in GF(q)$, we still have a parity check matrix of the same code. Due to this observation, we can look at the columns of the matrix as n points in the projective space $PG(d-1, q)$, given by their homogeneous coordinates. As shown in [1], if we start from an (R, μ) -MCF code, the point set S that we get this way, has the following properties:

- $R-1$ is the smallest number such that the $(R-1)$ -dimensional subspaces generated by R points of S cover the whole space $PG(d-1, q)$, and
- if a point X is not covered by any $(R-2)$ -dimensional subspace defined by S , then the number of $(R-1)$ -dimensional subspaces defined by R points of S containing X is at least μ (counted with multiplicity).

These structures are known under the name $(R-1, \mu)$ -saturating sets. One example is an oval in $PG(2, q)$ (q odd), which is a $(1, \frac{1}{2}(q-1))$ -saturating set of $q+1$ points. This means that there are at least $\frac{1}{2}(q-1)$ chords of the oval through any point not on the oval. The reader can find many other examples in [2].

3 Results and discussion

We will use the following notations:

- $s_q(d, R, \mu)$ is the minimum length of an (R, μ) -MCF code of codimension d .
- $s_q^*(d, R, \mu)$ is the minimum length of an (R, μ) -APMCF code of codimension d .
- $s_q^\sim(d, R, \mu)$ is the minimum length of an (R, μ) -AAPMCF code of codimension d .

Theorem 6 summarizes our main results.

- Theorem 6**
1. $s_q(d + d', R + R', \mu\mu') \leq s_q(d, R, \mu) + s_q(d', R', \mu')$
 2. $s_q^*(d + d', R + R', \mu\mu') \leq s_q^*(d, R, \mu) + s_q^*(d', R', \mu')$
 3. $s_q^\sim(d + d', R + R', \mu\mu') \leq s_q^\sim(d, R, \mu) + s_q^\sim(d', R', \mu')$

By induction on k we get the following:

- Collorary 7**
1. $s_q(\sum_{i=1}^k d_i, \sum_{i=1}^k R_i, \prod_{i=1}^k \mu_i) \leq \sum_{i=1}^k s_q(d_i, R_i, \mu_i)$
 2. $s_q^*(\sum_{i=1}^k d_i, \sum_{i=1}^k R_i, \prod_{i=1}^k \mu_i) \leq \sum_{i=1}^k s_q^*(d_i, R_i, \mu_i)$
 3. $s_q^\sim(\sum_{i=1}^k d_i, \sum_{i=1}^k R_i, \prod_{i=1}^k \mu_i) \leq \sum_{i=1}^k s_q^\sim(d_i, R_i, \mu_i)$

For the proof of Theorem 6 we need the following lemma from [1]:

Lemma 8 *Let C be a linear code of length n and codimension d and let $M \in GF(q)^{d \times n}$ be the parity check matrix of C . Then C is (R, μ) -MCF if and only if R is the smallest integer such that every $x \in GF(q)^d$ can be written as a linear combination of at most R columns of M and if x is not a linear combination of at most $R - 1$ columns, then it can be written as a linear combination of R columns in at least μ ways.*

Proof of Theorem 6: Let C be an (R, μ) -MCF code of length n and codimension d , and let C' be an (R', μ') -MCF code of length n' and codimension d' . Let us define a new code B as their direct sum:

$$B = C \oplus C' = \{cc' : c \in C, c' \in C'\}.$$

Clearly, the length of B is $n + n'$ and the dimension of B is $(n - d) + (n' - d') = (n + n') - (d + d')$, so the codimension of B is $d + d'$. We show that B is an $(R + R', \mu + \mu')$ -MCF code.

It is easy to see that the parity check matrix of B admits the form

$$N = \begin{pmatrix} M & 0 \\ 0 & M' \end{pmatrix},$$

where M is the parity check matrix of C and M' is the parity check matrix of C' . Suppose that $x \in GF(q)^{d+d'}$. Then it can be partitioned as $x = uv, u \in GF(q)^d, v \in GF(q)^{d'}$. By our hypotheses, u is a linear combination of at most R columns of M and v is a linear combination of at most R' columns of M' . Putting these columns together we

get at most $R + R'$ columns of N such that their sum is x . We still need that $R + R'$ is the smallest integer with this property, but this follows from the fact that R was smallest for C and R' was smallest for C' .

Furthermore, if x cannot be written as a linear combination of $R + R' - 1$ columns of N , then u is not a linear combination of $R - 1$ columns of M , and v is not a linear combination of $R' - 1$ columns of M' . This implies that there are at least μ ways to write u using columns of M , and there are at least μ' ways to write v using columns of M' . So we can combine these in at least $\mu\mu'$ ways. This completes the proof of the first statement of the theorem.

For the second and third part, we need the next claim, the proof of which is now omitted.

Claim 9

$$\gamma_{\mu\mu'}(C \oplus C') = \gamma_{\mu}(C)\gamma_{\mu'}(C').$$

This claim implies that if C and C' are both APMCF codes, then $C \oplus C'$ is also APMCF, and similarly, if C and C' are both AAPMCF codes, then $C \oplus C'$ is also AAPMCF, since $1 \cdot 1 = 1$.

We have seen that MCF codes correspond to multiple saturating sets in projective spaces. So what does the above direct sum construction look like in the geometric setting? Suppose that C is an $[n, n - d]_q$ code and C' is an $[n', n' - d']_q$ code. Let us denote the corresponding point sets in $PG(d - 1, q)$ and $PG(d' - 1, q)$ by $S(C)$ and $S(C')$, respectively. One can prove that the point set $S(C \oplus C')$ corresponding to the code $C \oplus C'$ can be obtained by taking a $(d - 1)$ -dimensional projective subspace U and a $(d' - 1)$ -dimensional projective subspace V in $PG(d + d' - 1, q)$ that are skew to each other, such that U contains $S(C)$ and V contains $S(C')$. Then $S(C \oplus C')$ is projectively equivalent the disjoint union of $S(C)$ and $S(C')$.

Acknowledgement

The author is thankful to György Kiss for all his help.

References

- [1] **D. Bartoli, A. A. Davydov, M. Giuletti, S. Marcugini, F. Pambianco**, Multiple coverings of the farthest-off points with small density from projective geometry, *Advan. Math. Commun.* **9** (1) (2015), 63-85.
- [2] **D. Bartoli, A. A. Davydov, M. Giuletti, S. Marcugini, F. Pambianco**, Further results on multiple coverings of the farthest-off point, *Advan. Math. Commun.* **10** (3) (2016), 613-632.

Section:

Combinatorics and Geometry

Organizer: Balázs Keszegh

Invited talk:

János Karl and **Géza Tóth**: Crossing lemma for the odd-crossing number

Contributions:

- Péter Ágoston: Semialgebraic sets as ranges of two-distance graphs
- Gábor Damásdi and Nóra Frankl: A note on convex geometric hypergraph
- Gábor Damásdi and Dömötör Pálvölgyi: A generalization of the Erdős-Sauer-Woodrow conjecture
- Rupert Levene and Narmada Varadarajan: Orthogonal projections for quantum channels and operator Systems



Crossing lemma for the odd-crossing number

János Karl and Géza Tóth

Budapest University of Technology and Economics

karlj@math.bme.hu, geza@renyi.hu

The *crossing number* of a graph G , $\text{CR}(G)$, is the minimum number of crossings (crossing points) over all drawings of G . The *pair-crossing number*, $\text{PCR}(G)$, is the minimum number of pairs of crossing edges over all drawings of G . In an optimal drawing for $\text{CR}(G)$, any two edges cross at most once. Therefore, it is not easy to see the difference between these two definitions. Indeed, there was some confusion in the literature between these two notions, until the systematic study of their relationship [PT00a]. Clearly, $\text{PCR}(G) \leq \text{CR}(G)$, and in fact, we cannot rule out the possibility, that $\text{CR}(G) = \text{PCR}(G)$ for every graph G . Probably it is the most interesting open problem in this area. From the other direction, the best known bound is $\text{CR}(G) = O(\text{PCR}(G)^{3/2} \log \text{PCR}(G))$ [KT21].

The *odd-crossing number*, $\text{OCR}(G)$, is the minimum number of pairs of edges that cross an odd number of times, over all drawings of G . Clearly, for every graph G , $\text{OCR}(G) \leq \text{PCR}(G) \leq \text{CR}(G)$. According to the (weak) Hanani-Tutte theorem [C34], [PSS07], if $\text{OCR}(G) = 0$, then G is planar, that is, $\text{OCR}(G) = \text{PCR}(G) = \text{CR}(G) = 0$. It was shown in [PSS07] that for $k = 1, 2, 3$, if $\text{OCR}(G) = k$, then $\text{OCR}(G) = \text{PCR}(G) = \text{CR}(G) = k$. There are examples where OCR is different from PCR and CR , there is an infinite family of graphs with $\text{OCR}(G) < 0.855 \cdot \text{PCR}(G)$ [T08], [PSS08]. From the other direction we only have $\text{PCR}(G) < 2\text{OCR}(G)^2$ [PT00a].

In [PT00b] some further variants were introduced, in order to study the role of crossings between adjacent edges. For each of CR , PCR , OCR , they introduced three counting rules:

Rule +: Only those drawings are considered, where adjacent edges cannot cross.

Rule 0: Adjacent edges can cross and their crossings are counted as well.

Rule -: Adjacent edges can cross and their crossings are not counted.

Combining these rules with the three crossing numbers, we get nine possibilities. But it is easy to see that $\text{CR}_+ = \text{CR}$ [PT00b]. On the other hand, regarding Rule + for the odd-crossing number, it seems more natural to assume that adjacent edges cross an *even number of times* than to assume that they do not cross at all. So, let $\text{OCR}_*(G)$ be the minimum number of odd-crossing pairs of edges over all drawings of G where adjacent edges cross an even number of times. Therefore, we have nine versions, see the table below. In this table, values do not decrease if we move to the right or up, and it was shown in [PSS08] that $\text{CR}(G) < 2\text{OCR}_-(G)^2$. On the other hand, there are graphs G , where $\text{OCR}_-(G) < \text{OCR}(G)$ [FPSS11].

Rule +	$\text{OCR}_*(G) \leq \text{OCR}_+(G)$	$\text{PCR}_+(G)$	$\text{CR}(G)$
Rule 0	$\text{OCR}(G)$	$\text{PCR}(G)$	
Rule -	$\text{OCR}_-(G)$	$\text{PCR}_-(G)$	$\text{CR}_-(G)$

The Crossing Lemma, discovered by Ajtai, Chvátal, Newborn, Szemerédi [ACNS82] and independently by Leighton [L84] is definitely the most important inequality for crossing numbers.

Crossing Lemma *If a simple graph G of n vertices has $m \geq 4.5n$ edges, then $\text{CR}(G) \geq \frac{1}{60.75} \frac{m^3}{n^2}$ edges.*

The bound is tight, apart from the value of the constant [PT97]. The constant above follows from the beautiful probabilistic argument of Chazelle, Sharir and Welzl [AZ04]. This argument works for all nine versions of the crossing number [PT00b]. For the classical crossing number, $\text{CR}(G)$, the constant was improved in three steps [PT97], [PRTT06], the best bound is due to Ackerman [A19], $\text{CR}(G) \geq \frac{1}{29} \frac{m^3}{n^2}$, when $m \geq 7n$.

The only improvement for any other version is a result of Ackerman and Schaefer [AS14], $\text{PCR}_+(G) \geq \frac{1}{34.2} \frac{m^3}{n^2}$, when $m \geq 6.75n$. For all other versions of the crossing number, the constant 60.75 is the best we have.

In this note we get an improvement for two other versions, OCR_+ and OCR_* .

Theorem 1 *Suppose that G has n vertices and $m \geq 6n$ edges. Then $\text{OCR}_+(G) \geq \text{OCR}_*(G) \geq \frac{1}{54} \frac{m^3}{n^2}$.*

Our approach is very similar to all previous improvements, mentioned above. The first step is to find many crossings in sparse graphs. Then this bound is applied for a random subgraph of G to get the general bound.

A graph G is called k -planar if it can be drawn in the plane such that there are at most k crossings on each edge. Such a drawing is called a k -plane drawing. Let $m_k(n)$ denote the maximum number of edges of a k -planar graph of n vertices.

Clearly, $m_0(n) = 3n - 6$. It is known that $m_1(n) = 4n - 8$ for $n \geq 12$, $m_2(n) \leq 5n - 10$ and it is tight for infinitely many values of n , [PT97], $m_3(n) \leq 5.5n - 11$, $m_4(n) \leq 6n - 12$, which are tight up to an additive constant [PRTT06], [A19].

We prove an odd-even version of these results. A graph G is called k -odd-planar if it can be drawn in the plane such that any edge is crossed *an odd number of times* by at most k other edges.

Let $m_k^{\text{odd}}(n)$ denote the maximum number of edges of a k -odd-planar graph. Such a drawing is called a k -odd-plane drawing. Clearly, we have $m_k^{\text{odd}}(n) \geq m_k(n)$ and by the weak Hanani-Tutte theorem [C34], [PSS07], we have $m_0^{\text{odd}}(n) = 3n - 6$.

Theorem 2 *For any $n, k \geq 1$ we have*

$$m_k^{\text{odd}}(n) \leq m_k(n) + k(n - 1).$$

We do not think that our bounds are tight. We cannot even rule out the possibility, that $m_k^{\text{odd}}(n) = m_k(n)$ for every n, k .

A (multi)graph G , together with its drawing D in the plane, is called *topological (multi)graph*. Let G be a topological multigraph, e an edge. The pieces of e in small neighborhoods of its endpoints are called *endings* of e and denoted by e^+ and e^- . The *rotation system* is the cyclic order of adjacent edges, or endings, at each vertex. A cyclic

order is always clockwise. Two edges form an *odd pair* (resp. *even pair*) if they cross an *odd* (resp. *even*) number of times. An edge is called *even* if it is crossed an even number of times by every other edge and it is *odd* otherwise.

According to the weak Hanani-Tutte theorem, if a graph can be drawn so that any two edges cross an even number of times, then it is planar. This result has many proofs, one of the nicest and simplest is due to Pelsmajer, Schaefer and Štefankovič [PSS07]. The proof is based on the following lemma:

Lemma 0. [PSS07] *Let G be a topological multigraph, which has one vertex and n edges (loops). Suppose that every edge is even. Then, G can be redrawn such that the rotation system is the same and there is no edge crossing.*

Theorem 2 is a consequence of the following generalization. We omit the proofs here.

Lemma 3 *Let G be a topological multigraph, which has one vertex and n edges (loops). Then, G can be redrawn such that (i) the rotation system is the same (ii) even pairs do not cross, (iii) odd pairs cross once, and (iv) there are no self-crossings.*

Remarks. 1. The statement of Lemma 3 can be found in [PSS07] as a remark. 2. Another possible proof is the following. Take a drawing of G which has the same rotation system and under this condition the minimum number of crossings. It can be shown that this drawing satisfies the conditions, but we did not find this method easier.

Proof of Theorem 1 We have $m_0(n) = 3n - 6$, and by Theorem 2, $m_1^{\text{odd}}(n) \leq m_1(n) + n - 1 = 5n - 9$. Therefore, it can be shown by induction on the number of edges that for any graph with n vertices and m edges, $\text{OCR}_*(G) \geq \text{OCR}(G) \geq 2m - 8n$.

Let G be a graph of n vertices and $m \geq 6n$ edges, drawn in the plane realizing $\text{OCR}_*(G)$, that is, any two adjacent edges cross an even number of times and there are $\text{OCR}_*(G)$ pairs of edges that cross an odd number of times. Take a random subgraph G' such that we take each vertex independently with probability $p = 6n/m$. Let n' , m' , and $x(G')$ denote the number of vertices (resp. edges) of G' , and the number of odd-crossing pairs of edges in G' , in the inherited drawing. We have $E(n') = pn$, $E(m') = p^2m$, $E(\text{OCR}_*(G')) \leq E(x(G')) = p^4\text{OCR}_*(G)$. For G' we have $\text{OCR}_*(G') \geq 2m' - 8n'$, therefore, $p^4\text{OCR}_*(G) \geq 2p^2m - 8pn$. Taking $p = 6n/m$, we get that $\text{OCR}_*(G) \geq \frac{1}{54} \frac{m^3}{n^2}$.

Remark. Combining Theorem 2 and the bounds for $m_k(n)$ we obtain that $m_1^{\text{odd}}(n) \leq 5n - 9$ and $m_2^{\text{odd}}(n) \leq 7n - 12$. In the proof of Theorem 1 we used only the first inequality, the second would not help. However, if we could prove that $m_2^{\text{odd}}(n) \leq 6.8n + c$ then we would get an improvement in Theorem 1 as well.

References

- [A19] E. Ackerman: On topological graphs with at most four crossings per edge, *Computational Geometry* **85** (2019), 101574.
- [AS14] E. Ackerman, M. Schaefer: A crossing lemma for the pair-crossing number, In: *Revised Selected Papers of the 22nd International Symposium on Graph Drawing*, (C. Duncan, A. Symvonis, eds.) *Lecture Notes in Computer Science*, **8871**, 222-233, Springer, Berlin, Heidelberg, 2014.

- [AZ04] M. Aigner, G. Ziegler: *Proofs from the Book*, Springer, Heidelberg, 2004.
- [ACNS82] M. Ajtai, V. Chvátal, M. Newborn, and E. Szemerédi, Crossing-free subgraphs, in: *Theory and Practice of Combinatorics, North-Holland Math. Stud.* **60**, North-Holland, Amsterdam-New York, 1982, 9-12.
- [C34] Ch. Chojnacki (H. Hanani): Über wesentlich unplättbare Kurven im dreidimensionalen Raume, *Fund. Math.* **23** (1934), 135-142.
- [FPSS11] R. Fulek, M. Pelsmajer, M. Schaefer, D. Štefankovič: Adjacent crossings do matter, In: *Revised Selected Papers of the 19nd International Symposium on Graph Drawing*, (van Kreveld, Speckmann, eds.) Lecture Notes in Computer Science, **7034**, 343-354, Springer, Berlin, Heidelberg, 2011.
- [KT21] J. Karl, G. Tóth, A slightly better bound on the crossing number in terms of the pair-crossing number. arxiv
- [L84] F. T. Leighton, New lower bound techniques for VLSI, *Math. Systems Theory* **17** (1984), 47-70.
- [LT79] R. J. Lipton, R. E. Tarjan: A separator theorem for planar graphs, *SIAM Journal on Applied Mathematics* **36** (1979), 177-189.
- [S17] M. Schaefer: *Crossing Numbers of Graphs*. CRC Press Published December 5, 2017. 350 Pages.
- [PRTT06] J. Pach, R. Radoičić, G. Tardos, G. Tóth: Improving the Crossing Lemma by finding more crossings in sparse graphs, *Discrete and Computational Geometry* **36**, (2006), 527-552.
- [PT97] J. Pach, G. Tóth: Graphs drawn with few crossings per edge, *Combinatorica* **17** (1997), 427-439.
- [PT00a] J. Pach, G. Tóth: Which crossing number is it anyway? *Journal of Combinatorial Theory, Series B* **80** (2000), 225-246.
- [PT00b] J. Pach, G. Tóth: Thirteen problems on crossing numbers, *Geombinatorics* **9** (2000), 194-207.
- [PSS07] M. Pelsmajer, M. Schaefer, D. Štefankovič: Removing even crossings, *Journal of Combinatorial Theory, Series B* **97** (2007), 489-500.
- [PSS08] M. Pelsmajer, M. Schaefer, D. Štefankovič: Odd crossing number and crossing number are not the same, *Discrete and Computational Geometry* **39** (2008), 442-454.
- [T08] G. Tóth: Note on the pair-crossing number and the odd-crossing number, *Discrete and Computational Geometry* **39**, (2008), 791-799.

Semialgebraic sets as ranges of two-distance graphs

Péter Ágoston

1 Introduction

We call a graph a *unit distance graph* (UDG) if it can be drawn in \mathbb{R}^2 so that the vertices are represented by distinct points and all neighbouring pairs of vertices have Euclidean distance 1. We call such a drawing a *unit distance representation* (UDR) of the graph. Unit distance graphs are a well-known notion in combinatorial geometry.[3, 5]

From now on, we suppose that all graphs are finite and simple unless stated otherwise.

We will be considering only finite and simple graphs. A graph G is an *edge-bicoloured graph* (EBG) if there is a fixed colouring of its edges with two colours, e.g. red and blue (denote the sets of red and blue edges of G by $E_r(G)$ and $E_b(G)$, respectively). Now we define a notion similar to UDGs: an EBG G is a $(1, d)$ -graph for some $d \in \mathbb{R}_{\geq 0}$ if there is an injective mapping of its vertices into the plane such that those connected with red, or blue edges go to points with distance 1, and d respectively. Such an embedding is called a $(1, d)$ -representation of G .

For an EBG G , define its range $\text{ran}(G)$ as the set of numbers for which G is a $(1, d)$ -graph. Let the range of a graph be the union of the ranges of its edge-bicolourings.

We call a graph with or without an edge-bicolouring a two-distance graph, if its range is not empty. Two-distance graphs have been studied in the past in several papers. [5][6]

Our main result is that (with respect to some boundedness criteria) the possible ranges of EBGs are exactly the semialgebraic subsets of \mathbb{R} (finite unions of intervals with algebraic endpoints).

First, some basic properties of the ranges of EBGs:

1. For EBGs $H \subseteq G$ (with the inherited colouring), $\text{ran}(G) \subseteq \text{ran}(H)$.
2. For EBGs G_1 and G_2 , we have $\text{ran}(G_1 \sqcup G_2) = \text{ran}(G_1) \cap \text{ran}(G_2)$.

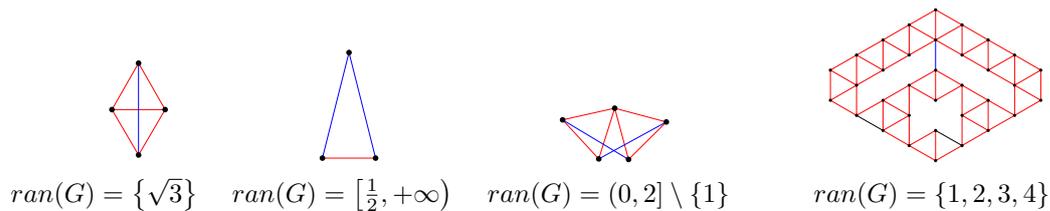


Figure 1: Four two-distance graphs and their ranges

3. For any EBG G , denote by G^* the EBG obtained by inverting the colouring ($E_b(G) = E_r(G^*)$ and $E_r(G) = E_b(G^*)$); then we get $\text{ran}(G^*) \cap \mathbb{R}_{>0} = \{d \in \mathbb{R}_{>0} \mid \frac{1}{d} \in \text{ran}(G)\}$.

And finally, a minor remark to show that it does not really make a difference whether we define the range of an EBG on $[0, +\infty)$ or on $(0, +\infty)$:

4. For any EBG G , $0 \in \text{ran}(G) \Leftrightarrow ((E_b(G) = \emptyset) \wedge (G \text{ is a UDG})) \Leftrightarrow \text{ran}(G) = [0, +\infty)$.

It is also known that deciding whether a number d is in the range of an EBG or not is \mathbb{R} -complete, since deciding whether a graph is a UDG or not is \mathbb{R} -complete. [7]

$\chi(\mathbb{R}^2)$ denotes the minimal number of colours needed to colour \mathbb{R}^2 without a monochromatic pair of distance 1. Finding $\chi(\mathbb{R}^2)$ is a famous problem [4] and Bukh conjectured [1] that by also forbidding a transcendental distance, we get the same number. If true, this could make it interesting to find graphs whose range only contains a transcendental number. But such graphs do not exist by Proposition 1 below.

Take the set of solutions (x_1, \dots, x_d) to a finite sequence of polynomial equations and inequalities of the form $p(x_1, \dots, x_d) = 0$ and $p(x_1, \dots, x_d) > 0$. If a set can be generated as the union of such sets, it is called a *semialgebraic set*. $S \subseteq \mathbb{R}$ is semialgebraic exactly if it can be obtained as the union of finitely many intervals with algebraic endpoints.

The Tarski–Seidenberg theorem [2, Theorem 1.5][has the following easy consequence as pointed out by Miklós Laczkovich (personal communication):

Proposition 1 *The range of an EBG G is always a semialgebraic set.*

Our main result says this condition is tight if $\text{ran}(G)$ has positive lower and upper bounds:

Theorem 1 *For a set $S \subseteq \mathbb{R}_{>0}$ with a positive lower and upper bound (λ and ν), there exists an EBG G with $\text{ran}(G) = S$ if and only if S is semialgebraic.*

2 Sketch of the proof of Theorem 1

Call a polynomial $p(x)$ even if all of its coefficients with odd index are 0, that is, if $p(x) = p(-x)$. Also, for any polynomial p and $0 < L \leq U < +\infty$, define

$$S_0(p, L, U) = \{x \in \mathbb{R}_{>0} \mid (p(x) \geq 0) \vee (x \leq L) \vee (x \geq U)\} \quad \text{and}$$

$$S_1(p, L, U) = \{x \in \mathbb{R}_{>0} \mid (p(x) > 0) \vee (x \leq L) \vee (x \geq U)\} \quad (\text{Figure 2}).$$

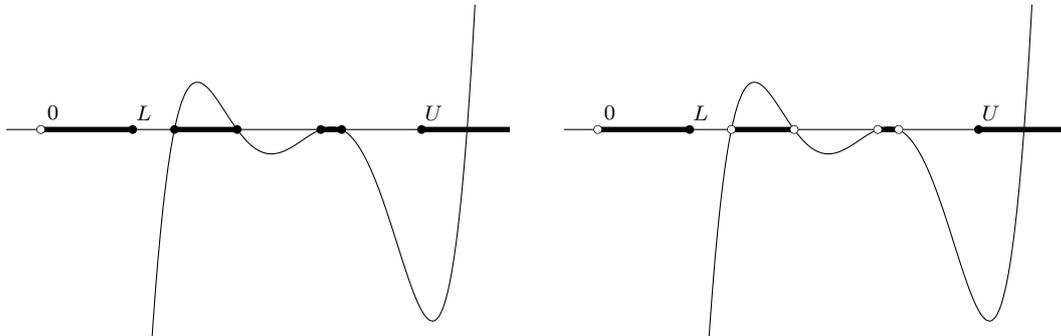


Figure 2: A polynomial $p(x)$ with $S_0(p, L, U)$ (left) and $S_1(p, L, U)$ (right) denoted by bold

Proposition 2 *For any even polynomial $p \in \mathbb{Z}[x]$ with integer coefficients and a negative leading coefficient, there exists an EBG $G(p)$, whose range is $S_0(p, 0, +\infty)$.*

Sketch of the proof: We define partly virtual EBGs (PVEBG), in which we also allow directed green edges, divided into groups. In a $(1, d)$ -representation of a PVEBG, we require green edges from the same group to have the same vector, besides the criteria for EBGs and we define its range analogously to EBGs. In case some boundedness conditions apply, an EBG with the same range can be created by replacing pairs of green edges from the same group by red grids (grids consisting only of red edges).

The most crucial component of creating $G(p)$ is graph A (Figure 3).

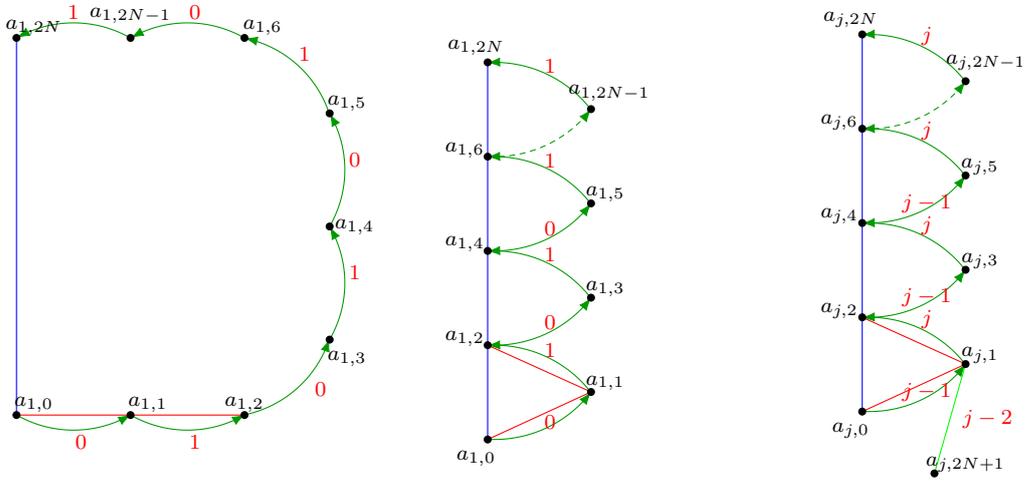


Figure 3: The components of A : A_1 (left), the only $(1, d)$ -representations (up to isometry) of A_1 (middle) and of A_j ($2 \leq j \leq \deg(p)$) (right) (N is large enough and groups are denoted by numbers).

For small enough d , A has exactly one $(1, d)$ -representation up to transformations which are isometries on the components. If we draw the complex plane so that $a_{1,0}\vec{a}_{1,1} = 1$ and $a_{1,1}\vec{a}_{1,2} = \varepsilon$, the members of the group marked by j will have vector ε^j and $|N \cdot (1 + \varepsilon)| = d$. This helps constructing points having distance of some even polynomial of d , and ultimately, constructing $G(p)$.

With a very small modification of $G(p)$, we get the following result:

Proposition 3 *For any even polynomial $p \in \mathbb{Z}[x]$ with a negative leading coefficient, there exists an EBG $G'(p)$, whose range is $S_1(p, 0, +\infty)$.*

And by a significant modification of any EBG G , we can prove the following proposition:

Proposition 4 *For an EBG G , positive rational numbers L_a, U_a and arbitrary real numbers L_b, U_b ($L_b < L_a < U_a < U_b$), if $\text{ran}(G) \cap (L, U) \neq \emptyset$, then there exists an EBG $G_{L_a, L_b}^{U_a, U_b}$ for which $\text{ran}\left(G_{L_a, L_b}^{U_a, U_b}\right) \cap (L_b, U_b) = ((0, L] \cup \text{ran}(G) \cup [U, +\infty)) \cap (L_b, U_b)$.*

Now we will use the following algebraic proposition:

Proposition 5 *Take a semialgebraic set $\sigma \subseteq [\lambda, \nu]$. For some $n \in \mathbb{N}$ there exist even polynomials p_1, \dots, p_{n+1} with integer coefficients and a negative leading coefficient, numbers $L_1, \dots, L_n, U_1, \dots, U_n \in \mathbb{Q}_{>0}$ and numbers $\zeta_1, \dots, \zeta_{n+1} \in \{0, 1\}$ so that*

$$\sigma = \left(\bigcap_{i=1}^n S_{\zeta_i}(p_i, L_i, U_i) \right) \cap S_{\zeta_{n+1}}(p_{n+1}, 0, +\infty).$$

Using the notations of Proposition 5, with the help of Proposition 4, we can construct $(1, d)$ -graphs $G(p_i)_{\lambda, L_i}^{v, U_i}$ for $1 \leq i \leq n$ and $\zeta_i = 0$, while in case of $\zeta_i = 1$, we construct $G'(p_i)_{\lambda, L_i}^{v, U_i}$, whose range coincides with $S_{\zeta_i}(p_i, L_i, U_i)$ on the interval $[\lambda, \nu]$. And finally, we can take $G'(p_{n+1})$, whose range is empty outside of $[\lambda, \nu]$, thus the intersection of the ranges of these graphs is σ because of Proposition 5. Thus their disjoint union has σ as its range. So all semialgebraic sets from $[\lambda, \nu]$ are the range of some EBG.

Now we will finish with a few open problems:

1. Can we construct such a graph for all semialgebraic sets (without the boundedness)?
2. What are the possible ranges of non-coloured graphs?
3. What about graphs coloured by more than two colours?
4. What is the situation in more than 2 dimensions?
5. What if we don't require the images of the vertices to be distinct?

Acknowledgement. I thank Dömötör Pálvölgyi for the problem and for helpful discussions and Miklós Laczkovich for proving Proposition 1.

References

- [1] B. BUKH: Measurable Sets With Excluded Distances, *GAF A Geom. funct. anal.* **18**, 668–697 (2008)
- [2] E. BIERSTONE, P. D. MILMAN: Semianalytic and subanalytic sets, *Inst. Hautes Études, Sci. Publ. Math.* **67** (1988), 5–42.
- [3] P. BRASS, W.O.J. MOSER, J. PACH: *Research Problems in Discrete Geometry* (2005)
- [4] A. D. N. J. DE GREY: The chromatic number of the plane is at least 5, arXiv:1804.02385 (2018)
- [5] G. EXOO, D. ISMAILESCU: A 6-chromatic two-distance graph in the plane, arXiv:1909.13177 (2019)
- [6] J. PARTS: A small 6-chromatic two-distance graph in the plane, arXiv:2010.12656 (2020)
- [7] M. SCHAEFER: Realizability of Graphs and Linkages, *Thirty Essays on Geometric Graph Theory* (2013), 461–482.

A note on convex geometric hypergraphs

Gábor Damásdi¹ and Nóra Frankl²

¹ ELTE, Budapest, ² Carnegie Mellon University, Pittsburgh and MIPT Moscow

damasdigabor@caesar.elte.hu, nfrankl@andrew.cmu.edu

1 Introduction

A *convex geometric hypergraph* (cgh for short) is a family of subsets of a set of points in general position in the plane. A *convex geometric 3-hypergraph* (3-cgh for short) is a 3-uniform convex geometric hypergraph. For a given configuration F of the edges, the extremal function $\text{ex}_\circlearrowleft(n, F)$ is the maximum number of edges in an F -free cgh on n vertices.

Two edges of a 3-cgh can determine eight different configurations regarding their intersection pattern. These configurations are shown in Figure 1, which is taken from [2].

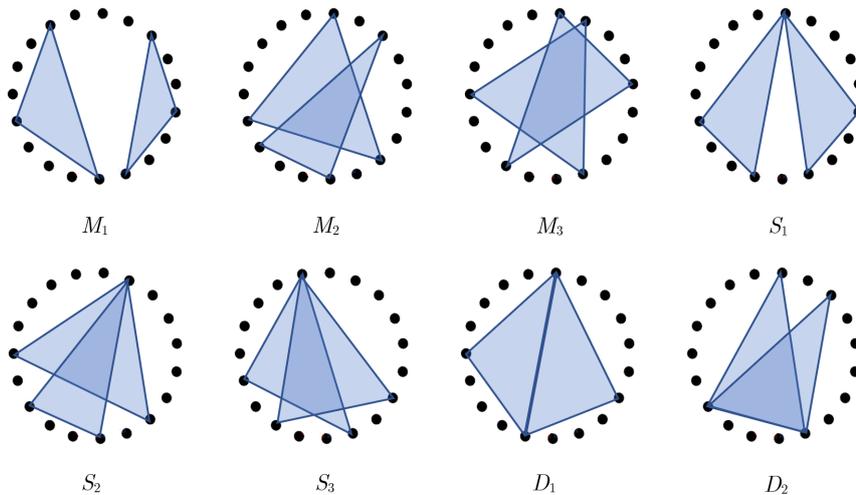


Figure 1: Intersection patterns of two triangles, taken from [2]

Braß[1] proved that the extremal function of any of these configurations is either $\Theta(n^2)$ or $\Theta(n^3)$. Füredi, Mubayi, O’Neill and Verstraëte [2] determined the extremal functions of five of these, namely of M_1, M_2, M_3 and D_1, S_1 exactly, and of S_3 asymptotically. They also proved that

$$\left\lfloor \frac{n^2}{4} \right\rfloor - 1 \leq \text{ex}_\circlearrowleft(n, S_2) \leq \frac{23n^2}{64}$$

and that

$$\frac{3}{14}n^2 - O(n) \leq \text{ex}_\circlearrowleft(n, D_2) \leq \frac{2n^2 - 3n}{9}.$$

The D_2 configuration consists of two triangles that share an edge and have a common interior point. We determine $\text{ex}_{\circlearrowleft}(n, D_2)$ exactly for all n , and exactly for $n \equiv 6 \pmod{9}$.

Theorem 1 *The maximum number of edges of a D_2 -free convex geometric 3-hypergraph on n vertices is $\frac{2}{9}n^2 + O(n)$. Moreover, if $n \equiv 6 \pmod{9}$, then the maximum possible number of edges is exactly $\frac{2n^2-3n}{9}$.*

Note that this matches the upper bound from [2]. Besides describing constructions with matching lower bounds, we also give two new proofs for the upper bound. We point out that finding the asymptotics for S_2 is still open, and in [2] it is conjectured that $\text{ex}_{\circlearrowleft}(n, S_2) = \lfloor n^2/4 \rfloor - 1$.

Our result can be generalised to r -uniform cgh's. Let $D_2(r)$ be the configuration formed by two convex r -gons that share an edge and have a common interior point. Then similarly to Theorem 1 we can show that $\text{ex}_{\circlearrowleft}(n, D_2(r)) = \frac{2n^2}{r^2} + O(n)$.

Convex geometric hypergraphs are closely related to ordered hypergraphs. There have been a lot of research done on both topics. For an overview, we refer to Braß[1], Pach [3, 4], and Tardos [5].

2 Proof of Theorem 1

Let \mathcal{H} be a 3-cgh on a set V of n vertices. Since we are only interested in the intersection pattern of the edges, we may assume that V is the vertex set of a regular convex n -gon. Assume that the edges are ordered triples (a, b, c) such that a, b, c are in this cyclic order according to the positive orientation.

We define the length $|\gamma|$ of an oriented arch γ as the number of vertices in the interior of γ plus one. (So that the length of an arch between consecutive vertices is 1.) The complement of every triangle (a, b, c) is the disjoint union of three open arcs $\gamma_{a,b}$, $\gamma_{b,c}$ and $\gamma_{c,a}$ such that $|\gamma_{a,b}| + |\gamma_{b,c}| + |\gamma_{c,a}| = n$. For every edge $(a, b, c) \in \mathcal{H}$ we associate the triple of arcs $(\gamma_{a,b}, \gamma_{b,c}, \gamma_{c,a})$ with the edge (a, b, c) . Observe that \mathcal{H} is D_2 -free if and only if for any two distinct edges $(a, b, c), (a', b', c')$ we have $\{\gamma_{a,b}, \gamma_{b,c}, \gamma_{c,a}\} \cap \{\gamma_{a',b'}, \gamma_{b',c'}, \gamma_{c',a'}\} = \emptyset$. In other words, \mathcal{H} is D_2 -free if and only if every open arc bounded by two vertices $a, b \in V$ with $a \neq b$ is associated with at most one edge. (Note that for every two vertices with $a \neq b$ there are two open arc bounded by a and b .)

Therefore, by finding $\frac{2}{9}n + O(1)$ pairwise disjoint triples of disjoint elements (x, y, z) such that $x, y, z \in \mathbb{N}$ and $x + y + z = n$, we can find $\frac{2}{9}n^2 + O(n)$ triangles without a D_2 configuration. Indeed, for each such triple (x, y, z) there are n triangles (a, b, c) such that $(|\gamma_{a,b}|, |\gamma_{b,c}|, |\gamma_{c,a}|) = (x, y, z)$. They are the cyclic shifts of a fixed triangle, as shown on Figure 2.

Thus, the following claim provides $\frac{2}{9}n^2 + O(n)$ triangles without a D_2 configuration.

Claim 2 *Suppose that $n = 9k + 6$. Then we can find $\frac{2}{9}n - \frac{1}{3}$ pairwise disjoint triples of disjoint elements (x, y, z) such that $x, y, z \in \mathbb{N}$ and $x + y + z = n$*

Proof: Consider the sets of triples

$$A = \{(i, 3k + 2 + i, 6k + 4 - 2i) \mid i \in \{1, \dots, k\}\}$$

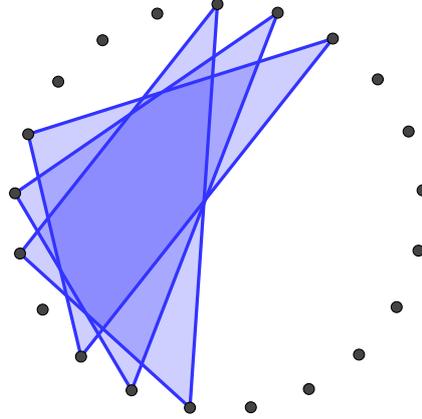


Figure 2: 3 cyclic shift of the triple $(4, 6, 11)$ with $n = 21$

and

$$B = \{(k + i, 2k + i + 1, 6k + 5 - 2i) | i \in \{1, \dots, k + 1\}\}.$$

Then $A \cup B$ contains $2k + 1$ triples. The sum of the numbers in each of these triples is $9k + 6 = n$. It is not hard to see that these triples are pairwise disjoint. Numbers from $[1, 2k + 1]$ appear in the first position of the triples, numbers from $[2k + 2, 4k + 2]$ appear in the second position and numbers from $[4k + 3, 6k + 3]$ in the third. Overall, we found $2k + 1 = \frac{2}{9}n - \frac{1}{3}$ disjoint triples. \square

By the the observation that every open arc bounded by two vertices $a, b \in V$ can be associated with at most one edge, the asymptotic upper bound on $\text{ex}_{\circlearrowleft}(n, D_2)$ follows from the claim below, for which we give two different proofs.

Claim 3 *Let M be the multiset that contains n copy of each integer $1 \leq i \leq n - 2$. Then M contains at most $\frac{2}{9}n^2 + 2n$ pairwise disjoint triples (x, y, z) for which $x + y + z = n$.*

Proof: Suppose we have found m triples. Let S be the sum of the numbers appearing in the triples with multiplicity. Since we have $3m$ numbers and each appears at most n times, the largest number appearing is at least $\lceil \frac{3m}{n} \rceil$. Therefor S is at least n times the sum of the first $\lfloor \frac{3m}{n} \rfloor$ positive integers. Hence, $S \geq n \binom{\lfloor \frac{3m}{n} \rfloor}{2}$. In each triple the sum of the numbers is n , thus $m \geq \frac{S}{n} \geq \binom{\lfloor \frac{3m}{n} \rfloor}{2} \geq \frac{1}{2}(\frac{9m^2}{n^2} - \frac{9m}{n})$. After rearranging we obtain $m \leq \frac{2}{9}n^2 + 2n$. \square

Proof: We define a fractional edge cover of the hypergraph of all triples $\{x, y, z\}$ with $x + y + z = n$ as follows. Let

$$w(i) := \frac{2}{3} - \frac{i}{n} \text{ if } i \leq \frac{2n}{3}$$

$$w(i) := 0 \text{ otherwise.}$$

For $n \equiv 6 \pmod{9}$ have $\sum_i w(i) = \frac{2}{9}n^2 - 3n$. By viewing the collection of disjoint triples as a matching, this implies the upper bound.

Note that the second proof gives sharp upper bound for $n \equiv 6 \pmod{9}$.

References

- [1] **Braß, P.**, Turán-type extremal problems for convex geometric hypergraphs, *Contemporary Mathematics*, 342:25–34, 2004
- [2] **Füredi, Z., Mubayi, D., O’Neill, J., and Verstraete, J.** Extremal problems for pairs of triangles in a convex polygon, *arXiv preprint: arXiv:2010.11100*, 2020.
- [3] **Pach, J.** Geometric graph theory, *Surveys in combinatorics 1999 (canterbury)*, 167–200. *London Math. Soc. Lecture Note Ser.*, 267.3
- [4] **Pach, J.** The beginnings of geometric graph theory. *In Erdős Centennial*, pages 465–484. *Springer*, 2013
- [5] **Tardos, G.** Extremal theory of ordered graphs. *Proceedings of the International Congress of Mathematicians: Rio de Janeiro 2018*. 2018.

A generalization of the Erdős-Sands-Sauer-Woodrow conjecture

Gábor Damásdi, Dömötör Pálvölgyi

MTA-ELTE Lendület Combinatorial Geometry Research Group

damasdigabor@caesar.elte.hu, dom@cs.elte.hu

The following question was asked by Sands, Sauer and Woodrow [1] and it is also due to Erdős:

Conjecture 1 (Erdős-Sands-Sauer-Woodrow) *For each n , is there a (least) positive integer $f(n)$ so that every finite tournament whose edges are coloured with n colours contains a set S of $f(n)$ vertices with the property that for every vertex u not in S there is a monochromatic path from u to a vertex of S ?*

The conjecture was recently solved by Bousquet, Lochet and Thomassé [2]. The ESSW conjecture has a number of nice applications, for example Gyárfás and Pálvölgyi showed that it implies the following result of Bárány and Lehel [3]. Every finite subset X of R^d can be covered by $f(d)$ X -boxes (i.e. each box has two antipodal points in X).

We show the following generalization of Conjecture 1.

Theorem 2 *For each n there is a (least) positive integer $f(n)$ so that every finite tournament whose edges are coloured with n colours contains a set S of $f(n)$ vertices with a partition $S = S_1 \cup \dots \cup S_n$ such that for every vertex u not in S there is an i such that there is a monochromatic path of color i from u to a vertex of S_i .*

1 Proof of Theorem 2.

The *closed in-neighbourhood* $N^-(x)$ of a vertex $x \in V$ is $\{x\} \cup \{y \mid (y, x) \in A\}$. Similarly the *closed out-neighbourhood* of a vertex x is $\{x\} \cup \{y \mid (x, y) \in A\}$. By extension, $N^-(S) = \cup_{x \in S} N^-(x)$ and $N^+(S) = \cup_{x \in S} N^+(x)$ when S is a subset of vertices. We say that a subset Q of the vertices is *dominated* if we have already found some vertices V such that there is a monochromatic path from each vertex of Q to some vertex of V . In this case V is called the *dominating set* of Q .

Let us quickly recap the main steps from the proof of Theorem 1 from [1]. First they carefully define a partition of the vertex set. For each part P they also define a probability distribution w_P on the vertices. Then, they dominate each part of the partition independently of the other parts using a probabilistic argument. Namely, they show that we can pick some points using w_P and they will dominate P with positive probability. Also, each part is dominated using edges of just one color. The only reason that the proof does not immediately work for Theorem 2 is that the dominating sets for the different parts might intersect.

The following proof of Theorem 2 follows a very similar path. We will also define a partition of the vertex set and corresponding probability distributions. Then, we will apply

the probabilistic argument for the parts simultaneously to ensure that the dominating sets of the parts are disjoint. To be able to ensure disjointness, the partition and the distributions have to be created a bit more carefully, we have to ensure that a vertex cannot have too much weight in any of the probability distributions.

The following useful lemma is from [4]:

Lemma 3 *If T is a tournament, then there exists a probability distribution w on $V(T)$ such that $w(N^-(x)) \geq 1/2$ for each $x \in V(T)$.*

The following lemma is a variant of Lemma 3.

Lemma 4 *Let T be a tournament and let $\delta > 0$ be a fixed number. If $|V(T)| > \frac{1}{\delta}$, then there exists a probability distribution w on $V(T)$ such that for each $x \in T$ one of the following holds.*

- $w(x) \leq 2\delta$ and $w(N^-(x)) \geq 1/2$
- $\delta \leq w(x) \leq 3\delta$

Note that the second condition holds for at most $1/\delta$ vertices.

The proof of Lemma 4 is based on an averaging argument. We repeatedly apply Lemma 3 and after each step we throw away the vertices whose weight is too much.

Lemma 5 *Let T be a complete multidigraph whose arc set is the union of k quasi-orders and let $\delta > 0$ be fixed. There exists a probability distribution w on $V(T)$ and a partition of $V(T)$ into sets T_1, T_2, \dots, T_k, D such that for every i and $x \in T_i$, we have $w(x) \leq 2\delta$ and $w(N_i^-(x)) \geq 1/2k$ and for every $x \in D$ we have $\delta \leq w(x) \leq 3\delta$.*

Proof: Take w according to Lemma 4. For every i in $[k]$, let T_i be the subset of vertices such that $w(N_i^-(x)) \geq 1/2k$. The sets T_i cover those vertices that satisfy the first property in Lemma 4, so we can extract a partition with the required properties.

Now we are ready to tackle Theorem 2.

Proof: Fix $0 < \delta < 1$. We will choose its value later. Let $\mathcal{I} = \{\emptyset\} \cup \bigcup_{l=1}^{k+1} [k]^l$, that is, sequences of length at most $k+1$ whose terms are from $[k]$. Now for a tournament T we are going to define a system of subsets of the vertex set $\{T_{j_1, \dots, j_i}\}_{(j_1, \dots, j_i) \in \text{Ind}}$ for some index set $\text{Ind} \subset \mathcal{I}$. The process that defines these subsets will be similar to a tree traversal algorithm. For each T_{j_1, \dots, j_i} we will also define a probability distribution w_{j_1, \dots, j_i} which is concentrated on T_{j_1, \dots, j_i} .

We start by setting $T_\emptyset = T$ and we apply Lemma 5 for T_\emptyset to obtain T_1, T_2, \dots, T_k and D_\emptyset together with a probability distribution w_\emptyset . That is, for every i and $x \in T_i$, we have $w_\emptyset(x) \leq 2\delta$ and $w_\emptyset(N_i^-(x)) \geq 1/2k$.

Then, as long as possible, we do the following step. We pick a previously not selected T_{j_1, \dots, j_i} from the already defined ones such that j_1, \dots, j_i are pairwise distinct, and $|T_{j_1, \dots, j_i}| > \frac{1}{\delta}$. For such a T_{j_1, \dots, j_i} we apply Lemma 5 to obtain the partition $T_{j_1, \dots, j_i, 1}, \dots, T_{j_1, \dots, j_i, k}, D_{j_1, \dots, j_i}$ and a probability distribution w_{j_1, \dots, j_i} .

From Lemma 4 we know the following:

- $|D_{j_1, \dots, j_i}| \leq \frac{1}{\delta}$

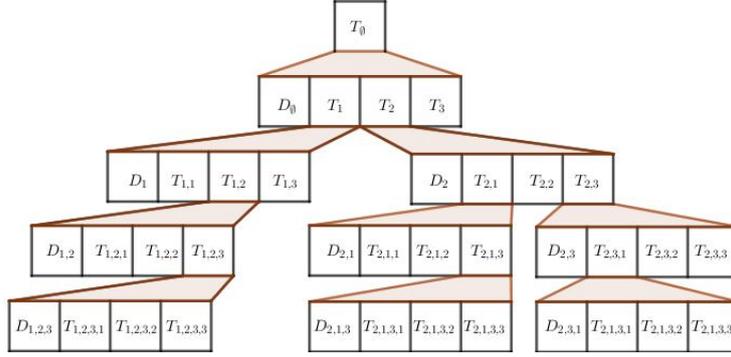


Figure 1: The outcome of the partition process for $k = 3$ assuming that the size of T_3 and $T_{1,3}$ is less than $1/\delta$.

- For every $l \in [k]$ and $x \in T_{j_1, \dots, j_i, l}$ we have $w_{j_1, \dots, j_i}(N_l^-(x)) \geq 1/2k$
- $w_{j_1, \dots, j_i}(x) \leq 3\delta$ for every $x \in V(T)$

This process terminates in (strictly) less than $|\mathcal{I}| \leq k^{k+2}$ steps as no index sequence can be longer than $k + 1$.

When the process halts we define a partition of the vertex set by taking every set of the following three kind. D_{j_1, \dots, j_i} -s, T_{j_1, \dots, j_i} -s that have fewer elements than $\frac{1}{\delta}$ and T_{j_1, \dots, j_i} -s where $j_i = j_l$ for some $l < i$. These sets are clearly disjoint, and they cover the vertex set, since the process halted. Let D denote the union of the D_{j_1, \dots, j_i} -s, let T_{small} denote the union the T_{j_1, \dots, j_i} -s that have fewer elements than $1/\delta$ and finally let T_{rep} denote the union of the T_{j_1, \dots, j_i} -s where $j_i = j_l$ for some $l < i$.

We dominate each part of the partition. Since each D_{j_1, \dots, j_i} has fewer elements than $\frac{1}{\delta}$ we have $|D| \leq \frac{1}{\delta} k^{k+2}$. Similarly $|T_{small}| \leq \frac{1}{\delta} k^{k+2}$. We will simply put each vertex from $D \cup T_{small}$ into our dominating set of the first color (S_1) but we will not use them to dominate any point other than themselves. So, it remains to dominate the vertices in T_{rep} .

Let $I_{rep} \subset \mathcal{I}$ be the index set of those T_{j_1, \dots, j_i} -s where $j_i = j_l$ for some $l < i$. Let $(j_1, \dots, j_i) \in I_{rep}$ and consider T_{j_1, \dots, j_i} , $T_{j_1, \dots, j_{i-1}}$ and $T_{j_1, \dots, j_{i-1}}$. Note that for each $x \in T_{j_1, \dots, j_i}$ we have $w_{j_1, \dots, j_{i-1}}(N_{j_i}^-(x)) \geq \frac{1}{2k}$. Hence, if we can find $S_{j_1, \dots, j_i} \subset V(T)$ such that $w_{j_1, \dots, j_{i-1}}(N_{j_i}^+(S_{j_1, \dots, j_i})) > 1 - \frac{1}{2k}$, then T_{j_1, \dots, j_i} is dominated by S_{j_1, \dots, j_i} . We will pick the S_{j_1, \dots, j_i} -s randomly and we will show that the probability of $w_{j_1, \dots, j_{i-1}}(N_{j_i}^+(S_{j_1, \dots, j_i})) > 1 - \frac{1}{2k}$ for each $(j_1, \dots, j_i) \in I_{rep}$ and having all the S_{j_1, \dots, j_i} -s pairwise disjoint is positive.

Let $0 < \varepsilon < 1$ be fixed, we will choose its value later. Let $g(\varepsilon) = \lfloor \frac{\ln(\varepsilon)}{\ln(1 - \frac{1}{2k})} \rfloor + 1$ and let S_{j_1, \dots, j_i} be a multiset of $g(\varepsilon)$ elements picked independently at random according to the distribution $w_{j_1, \dots, j_{i-1}}$. For every vertex $x \in T_{j_1, \dots, j_{i-1}}$, $P(x \in N_{j_i}^+(S)) \geq 1 - (1 - \frac{1}{2k})^{g(\varepsilon)} \geq 1 - \varepsilon$. Therefore, by linearity of expectation,

$$\mathbb{E}(w_{j_1, \dots, j_{i-1}}(N_{j_i}^+(S_{j_1, \dots, j_i}))) \geq \sum_{x \in V(T)} w_{j_1, \dots, j_{i-1}}(x) \cdot (1 - \varepsilon) \geq 1 - \varepsilon.$$

Let X be the random variable $\sum_{(j_1, \dots, j_i) \in I_{rep}} w_{j_1, \dots, j_{i-1}}(N_{j_i}^+(S_{j_1, \dots, j_i}))$. Clearly,

$$\mathbb{E}(X) \geq |I_{rep}|(1 - \varepsilon).$$

Let Y be the indicator variable of the event that the multiset $\bigcup_{(j_1, \dots, j_i) \in I_{rep}} S_{j_1, \dots, j_i}$ does not contain a vertex of multiplicity more than one. Let us consider the random variable $X \cdot Y$. If we can show that there is an event when $X \cdot Y > |I_{rep}| - \frac{1}{2k}$, then we are done. This would imply that $Y = 1$, that is the S_{j_1, \dots, j_i} -s are disjoint. Since $w_{j_1, \dots, j_{i-1}}(N_{j_i}^+(S_{j_1, \dots, j_i})) \leq 1$ for each $(j_1, \dots, j_i) \in I_{rep}$, it would also mean that $w_{j_1, \dots, j_{i-1}}(N_{j_i}^+(S_{j_1, \dots, j_i})) > 1 - \frac{1}{2k}$ for each $(j_1, \dots, j_i) \in I_{rep}$.

The rest of the proof is a standard probabilistic calculation. With the right choice of ε and δ we can show that $\mathbb{E}(XY) \geq |I_{rep}| - \frac{1}{2k}$. Hence, there is an event when $X \cdot Y > |I_{rep}| - \frac{1}{2k}$, finishing the proof.

2 Final remarks

The main motivation behind Theorem 2 was that it can be applied for geometric hypergraph coloring problems. Using Theorem 2 it can be proved that there is a n_0 such that for any finite point set P in the plane and any convex set C the points of P can be three-colored such that there is no translate of C containing at least n_0 points of P , all of the same color.

References

- [1] **Bousquet, Nicolas and Lochet, William and Thomassé, Stéphan** A proof of the Erdos-Sands-Sauer-Woodrow conjecture, *J. Combin. Theory Ser. B*, **137** (2019), 316–319.
- [2] **Sands, B. and Sauer, N. and Woodrow, R.** On monochromatic paths in edge-coloured digraphs, *J. Combin. Theory Ser. B*, **137** (1982), 271–275.
- [3] **I. Bárány and J. Lehel** Covering with Euclidean boxes, *European Journal of Combinatorics*, **8**, (1987), 113–119.
- [4] **D. Fisher and J. Ryan** Probabilities within optimal strategies for tournament games. *Discrete Applied Math.*, **56**, (1995), 87–91.

Orthogonal Projections for Quantum Channels and Operator Systems

Rupert Levene, Narmada Varadarajan

Department of Mathematics & Statistics, University College Dublin

narmada.varadarajan@ucdconnect.ie

Abstract

In quantum information theory, a quantum graph or operator system plays the role of the confusability graph from classical information theory. Classical graph parameters extend to operator systems by letting projections play the role of vertices, and orthogonality the role of non-adjacency. We review the definition of connected operator systems, and define the number of connected components for disconnected operator systems. This extends the classical graph-theoretic equivalent. We also show how these methods help to study the independence number of a quantum graph.

1 Introduction

1.1 Classical information theory

In classical information theory, a *noisy channel* N from an input alphabet X to an output alphabet Y is a function that sends each input from X to a probability distribution on Y .

$$N : x \rightarrow \left(p(y|x) \right)_{y \in Y}.$$

An equivalent model is as a linear map $\mathcal{N} : \mathbb{C}^X \rightarrow \mathbb{C}^Y$ that sends each basis vector of \mathbb{C}^X to a probability distribution vector in \mathbb{C}^Y . That is,

$$\mathcal{N}(x) = \left(p(y|x) \right)_{y \in Y}.$$

For a fixed choice of basis, noisy channels from $X \rightarrow Y$ are in one-to-one correspondence with matrices $\mathbb{C}^X \rightarrow \mathbb{C}^Y$ that map probability distributions to probability distributions.

One is typically interested in the accuracy of the information received from a noisy channel. To this end, Shannon [3] initiated the study of *zero-error capacities* by introducing the *confusability graph* of a channel. The *confusability graph* G of a noisy channel $N : X \rightarrow Y$ is the graph on vertex set X with edge set

$$E(G) = \left\{ \{x_i, x_j\} : x_i \neq x_j \text{ and } \exists y \in Y, p(y|x_i) \cdot p(y|x_j) \neq 0 \right\}.$$

Intuitively, two vertices of X are connected by an edge if they can be “confused” by the channel. By construction, the confusability graph of a channel on X is a simple graph on the vertex set. Conversely, given a simple graph G on vertex set X , there exists an output alphabet Y and a channel $N : X \rightarrow Y$ so that G is the confusability graph of N . However,

the alphabet Y and the channel N are not necessarily unique; several channels can give rise to the same confusability graph.

Shannon showed that various measures of channel capacity depend only on the confusability graphs. One such measure of interest is the independence number of a graph, which indicates the maximum number of input letters that can be transmitted by the channel without “confusion”.

1.2 Quantum information theory

While a classical channel is a linear map between vector spaces, a quantum channel is a linear map between matrix algebras.

Definition 1 *A quantum channel is a linear map $\phi : M_n \rightarrow M_d$ that satisfies the following.*

- (i) *It is completely positive: It has a Kraus decomposition $\phi(X) = \sum_{i=1}^r E_i X E_i^*$, for some matrices (called Kraus operators) $E_1, \dots, E_r \in M_{d,n}$.*
- (ii) *It is trace-preserving: $\sum_{i=1}^r E_i^* E_i = I_n \in M_n$.*

Duan, Severini, and Winter [2] introduced the *quantum confusability graph* as an analog of the classical case.

Definition 2 [2] *The quantum confusability graph of a quantum channel $\phi : M_n \rightarrow M_d$ with Kraus decomposition $\phi(X) = \sum_{i=1}^r E_i X E_i^*$ is the set*

$$\mathcal{S} = \text{span}\{E_i^* E_j : i, j = 1, \dots, r\}.$$

Although the Kraus operators of a quantum channel are not unique, the confusability graph is independent of the choice of Kraus operators. An *operator system* in M_n is a linear subspace closed under adjoints and containing the identity. It is clear from the definition that every quantum confusability graph is an operator system. The converse correspondence is well-known.

Claim 3 *Every operator system in M_n arises as the confusability graph of some quantum channel.*

Just as in the classical case, given an operator system \mathcal{S} , its associated channel ϕ and output algebra M_d are not unique; different channels can give rise to the same operator system.

2 Extending connectivity to quantum graphs

Given a classical graph G on n vertices, its associated operator system in M_n is defined as

$$\mathcal{S}_G = \{E_{i,j} : i = j, \text{ or } ij \in E(G)\}.$$

Several classical graph parameters like the independence number or chromatic number can be generalised to operator systems. In [1], the authors define *connectivity* for operator systems.

Definition 4 [1] *An operator system $\mathcal{S} \subset M_n$ is said to be connected if one of the following equivalent conditions holds.*

- (1) *For every nontrivial projection $P \in M_n$, $P\mathcal{S}(I_n - P) \neq \{0\}$.*
- (2) *For some $m \in \mathbb{N}$, $\mathcal{S}^m = M_n$, where $\mathcal{S}^m = \text{span}\{A_1 \cdots A_m : A_i \in \mathcal{S}\}$.*

The first condition most closely mirrors the classical definition; a classical graph is connected if every nontrivial partition of the vertex set has a crossing edge. While the results in [1] focus on connected operator systems, our work focuses on disconnected operator systems.

2.1 Connected components

Since M_n is finite-dimensional, the chain $(\mathcal{S}^m)_{m \in \mathbb{N}}$ has to terminate after finitely many steps at $C^*(\mathcal{S})$, the C^* -algebra generated by \mathcal{S} . Thus,

Definition 5 *An operator system $\mathcal{S} \subset M_n$ is disconnected if one of the following equivalent conditions holds.*

- (1) *For some nontrivial projection $P \in M_n$, $P\mathcal{S}(I_n - P) = \{0\}$.*
- (2) *$C^*(\mathcal{S})$ is a proper subalgebra of M_n .*

The first condition is what enables us to define the number of connected components for an operator system:

Definition 6 *The number of connected components of an operator system \mathcal{S} is the maximal integer l for which there exist projections P_1, \dots, P_l such that*

- (i) $P_1 + \cdots + P_l = I_n$, and
- (ii) $P_i \mathcal{S} P_j = \{0\}$ for all $i \neq j$.

We denote this number by $\delta(\mathcal{S})$.

In other words, every matrix of \mathcal{S} has a block decomposition with respect to the projections $\{P_1, \dots, P_l\}$. Since block decompositions are preserved under matrix products,

Lemma 7 *For any operator system \mathcal{S} , $\delta(\mathcal{S}) = \delta(C^*(\mathcal{S}))$.*

Using the representation theory of finite-dimensional C^* -algebras, we can explicitly compute the number of connected components.

Theorem 8 *Every finite-dimensional C^* -algebra \mathcal{A} is of the form*

$$\mathcal{A} \cong \bigoplus_{i=1}^k M_{n_i} \otimes I_{m_i}.$$

Our first main result is

Theorem 9 *Let \mathcal{S} be an operator system and $\mathcal{A} = C^*(\mathcal{S})$. If \mathcal{A} has a C^* -decomposition as above, then*

$$\delta(\mathcal{S}) = \delta(\mathcal{A}) = \sum_{i=1}^k m_i.$$

Further, any maximal disconnecting family is of the form

$$\bigcup_{i=1}^k \left\{ I_{n_i} \otimes E_j : E_j \in M_{m_i} \text{ is a rank-1 projection, } j = 1, \dots, m_i \right\}$$

Intuitively, any maximal block decomposition of \mathcal{A} has to be compatible with its C^* -algebra decomposition.

3 Further work

The language of C^* -algebras promises to hold interesting applications to quantum channels. For example, the number of connected components gives us a (sometimes trivial) lower bound on the independence number. The *independence number* of an operator system \mathcal{S} , $\alpha(\mathcal{S})$, is the maximal number of projections $\{P_1, \dots, P_l\}$ such that $P_i \mathcal{S} P_j = \{0\}$ for all $i \neq j$. This closely resembles $\delta(\mathcal{S})$, only we do not require the projections to sum to the identity.

Corollary 10 *For any operator system \mathcal{S} , $\alpha(\mathcal{S}) \geq \delta(\mathcal{S})$.*

With a little manipulation, one can obtain an exact result for C^* -algebras.

Corollary 11 *For any C^* -algebra \mathcal{A} , $\alpha(\mathcal{A}) = \delta(\mathcal{A})$.*

References

- [1] **Chávez-Domínguez, A. and Swift, A.** Connectivity for Quantum Graphs, *Linear Algebra and its Applications*, **608** (2021), 37–53.
- [2] **Duan, R., Severini, S. and Winter, A.,** Zero-Error Communication via Quantum Channels, Noncommutative Graphs, and a Quantum Lovász Number, *IEEE Transactions on Information Theory*, **59** (2013), 1164–1174.
- [3] **Shannon, C.,** The zero error capacity of a noisy channel, *IRE Transactions on Information Theory*, **2** (1956), 8–19.

Section:

Geometric constraint systems: theory and algorithms

Organizer: Tibor Jordán

Invited talk:

Bill Jackson, **Viktória E. Kaszanitzky** and Bernd Schulze: Scene analysis with symmetry

Contributions:

- Dániel Garamvölgyi: Algebraic matroids and global rigidity
- Tibor Jordán: Rigid block and hole graphs with a single block
- Csaba Király and András Mihálykó: Localizable sensor networks with optimal anchor sets I.: A min-max theorem
- Csaba Király and András Mihálykó: Localizable sensor networks with optimal anchor sets II.: An algorithm

Scene analysis with symmetry

Bill Jackson

School of Mathematical Sciences, Queen Mary University of London

b.jackson@qmul.ac.uk

Viktória E. Kaszanitzky

Department of Computer Science and Information Theory, Budapest University of
Technology and Economics

kaszanitzky@cs.bme.hu

Bernd Schulze

Department of Mathematics and Statistics, Lancaster University

b.schulze@lancaster.ac.uk

1 Introduction

Given an incidence structure S and a straight line drawing of S in the plane, one may ask whether this drawing is the vertical projection of a spatial polyhedral scene. This is a well studied question in Discrete Geometry which has some beautiful connections to areas such as Geometric Rigidity Theory and Polytope Theory, see [5] for details. Moreover, this problem has important applications in Artificial Intelligence, Computer Vision and Robotics. In this paper we consider *symmetric* drawings and their vertical lifting properties.

1.1 Basic definitions and results

A (*polyhedral*) *incidence structure* S is an abstract set of vertices V , an abstract set of faces F , and a set of incidences $I \subseteq V \times F$.

A $(d-1)$ -*picture* is an incidence structure S together with a corresponding location map $r : V \rightarrow \mathbb{R}^{d-1}$, and is denoted by $S(r)$. A d -*scene* $S(p, P)$ is an incidence structure $S = (V, F; I)$ together with a pair of location maps, $p : V \rightarrow \mathbb{R}^d$, and $P : F \rightarrow \mathbb{R}^d$, such that for each face F_j the vertices incident with F_j lie in a hyperplane. (Here P is an assignment of normal vectors to the faces.) A *lifting* of a $(d-1)$ -picture $S(r)$ is a d -scene $S(p, P)$, with the vertical projection $\Pi(p) = r$.

A lifting $S(p, P)$ is *trivial* if all the faces lie in the same hyperplane. Further, $S(p, P)$ is *folded* (or *non-trivial*) if some pair of faces lie in different hyperplanes, and is *sharp* if each pair of faces sharing a vertex lie in distinct hyperplanes. A picture is called *sharp* if it has a sharp lifting. Moreover, a picture which has no non-trivial lifting is called *flat* (or *trivial*). A picture with a non-trivial lifting is called *foldable*.

Theorem 1 (Picture Theorem) [4],[5] *A generic $(d-1)$ -picture of an incidence structure $S = (V, F; I)$ with at least two faces has a sharp lifting, unique up to lifting equivalence,*

if and only if $|I| = |V| + d|F| - (d + 1)$ and $|I'| \leq |V'| + d|F'| - (d + 1)$ for all subsets I' of incidences with at least two faces.

The lifting matrix of a generic $(d - 1)$ -picture S has independent rows if and only if for all non-empty subsets I' of incidences, we have $|I'| \leq |V'| + d|F'| - d$.

1.2 Symmetric incidence structures and pictures

An *automorphism* of an incidence structure $S = (V, F; I)$ is a pair $\alpha = (\pi, \sigma)$, where π is a permutation of V and σ is a permutation of F such that $(v, f) \in I$ if and only if $(\pi(v), \sigma(f)) \in I$ for all $v \in V$ and $f \in F$. For simplicity, we will write $\alpha(v)$ for $\pi(v)$ and $\alpha(f)$ for $\sigma(f)$.

The automorphisms of S form a group under composition, denoted $\text{Aut}(S)$. An *action* of a group Γ on S is a group homomorphism $\theta : \Gamma \rightarrow \text{Aut}(S)$. The incidence structure S is called Γ -*symmetric* (with respect to θ) if there is such an action.

Let Γ be an abstract group, and let S be a Γ -symmetric incidence structure (with respect to θ). Further, suppose there exists a group representation $\tau : \Gamma \rightarrow O(\mathbb{R}^{d-1})$. Then we say that a picture $S(r)$ is Γ -*symmetric* (with respect to θ and τ) if

$$\tau(\gamma)(r_i) = r_{\theta(\gamma)(i)} \text{ for all } i \in V \text{ and all } \gamma \in \Gamma. \quad (1)$$

In this case we also say that $\tau(\Gamma) = \{\tau(\gamma) \mid \gamma \in \Gamma\}$ is a *symmetry group* of $S(r)$.

A symmetric picture is called $\tau(\Gamma)$ -*generic* if the vertex positions are "as generic as possible", that is, the only correspondence among the coordinates of the vertices is implied by the symmetry group $\tau(\Gamma)$.

2 Liftings with incidental symmetry

Now we summarise results regarding the effect of symmetry on the lifting properties of $(d - 1)$ -pictures. It was proven in [1] that the number of vertices, faces and incidences fixed by the elements of Γ play a key role in the foldability of symmetric pictures. For every symmetry group of the plane a necessary condition for minimal flatness was given.

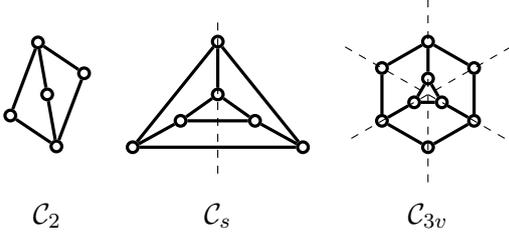


Figure 1: Some symmetric 2-pictures with a (sharp) symmetry-induced lifting with 2-fold rotational, reflectional and dihedral symmetry (where all interior regions are faces). All of these structures are flat in a generic non-symmetric position.

In the next two results \mathcal{C}_3 is the 3-fold rotational group and V_3 and I_3 denote the set of vertices and incidences fixed by the 3-fold rotation, see [1] for a detailed definition.

Theorem 2 [2] *A \mathcal{C}_3 -symmetric incidence structure $S = (V, F; I)$ is \mathcal{C}_3 -generically minimally flat if and only if $|I| = |V| + 3|F| - 3$, $|I'| \leq |V'| + 3|F'| - 3$ for every subset of incidences $|I'|$ with at least one face and $|I_3(S)| = |V_3(S)|$.*

Theorem 3 [2] *Let $S = (V, F, I)$ be a \mathcal{C}_3 -symmetric incidence structure with $|I'| \leq |V'| + 3|F'| - 4$ for every substructure of S with at least two faces.*

1. If $|V_3(S)| = 0$ then S is \mathcal{C}_3 -generically sharp.
2. If $|V_3(S)| = |I_3(S)| = 1$ and $|I'| \leq |V'| + 3|F'| - 6$ holds for every \mathcal{C}_3 -symmetric substructure of S with at least two faces, then S is \mathcal{C}_3 -generically sharp.

3 Liftings with forced symmetry

In this section we consider the case where the resulting d -scene is required to "extend" the symmetry into a higher dimension.

We first give an example of a symmetric $(d - 1)$ -picture that is foldable, but none of its folded liftings "extends" the symmetry of the $(d - 1)$ -picture. Consider the 2-picture in Figure 2. Using Theorem 1 it is easy to see that this 2-picture has a non-trivial lifting as it does not have enough incidences to be flat since $|I| = |V| + 3|F| - 4 = 16$. On the other hand consider a lifting of the same 2-picture which admits a 4-fold rotational symmetry around the z -axis. Such a symmetry forces the vertices belonging to the same vertex orbit to lie in a plane orthogonal to the z -axis. But then the constraints corresponding to the faces force every vertex to lie in the same plane, so the 3-scene must be flat.

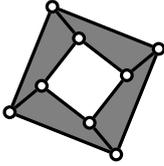


Figure 2: A 2-picture with 4-fold rotational symmetry around the origin that has a non-trivial lifting but has no non-trivial symmetric lifting which admits 4-fold rotational symmetry around the z axis. The 2-scene consists of 8 vertices which belong to two vertex orbits and four faces (shown in gray colour) which belong to the same face orbit.

3.1 Formal definitions

Let $S(r)$ be a Γ -symmetric $(d - 1)$ -picture with symmetry group $\tau(\Gamma)$ and let $\tau' : \Gamma \rightarrow O(\mathbb{R}^d)$ be a representation of Γ so that:

1. the hyperplane of $S(r)$ is invariant under $\tau'(\Gamma)$;
2. the restriction of $\tau'(\Gamma)$ to the hyperplane of $S(r)$ is $\tau(\Gamma)$.

We say that $S(r)$ is $\tau'(\Gamma)$ -*symmetry-forced flat* if it has no non-trivial $\tau'(\Gamma)$ -symmetric liftings. Otherwise it is $\tau'(\Gamma)$ -*symmetry-forced foldable*. If it has a $\tau'(\Gamma)$ -symmetric sharp lifting then it is $\tau'(\Gamma)$ -*symmetry-forced sharp*.

In order to state our results we also need to define a quotient incidence structure. We choose a set of representatives $\mathcal{O}_V = \{v_1, \dots, v_n\}$, one for each vertex orbit. Similarly, let $\mathcal{O}_F = \{f_1, \dots, f_m\}$ and $\mathcal{O}_I = \{i_1, \dots, i_k\}$ be the sets of representatives of F and I , respectively. If $i_l = (\gamma_1 v_i, \gamma_2 f_j) \in I$ where $i_l \in \mathcal{O}_I$, $v_i \in \mathcal{O}_V$, $f_j \in \mathcal{O}_F$ and $\gamma_1, \gamma_2 \in \Gamma$ then we assign $\gamma_1^{-1} \gamma_2$ to i_l . We will use the notation $\psi(i_l) = \gamma_1^{-1} \gamma_2$.

The *gain bipartite graph* (G_S, ψ) of a Γ -symmetric incidence structure S is an edge-labeled bipartite directed multigraph constructed as follows. The two vertex classes are \mathcal{O}_V and \mathcal{O}_F and there is an edge with label γ between v_i and f_j for each possible group element γ for which $i_l = (v_i, \gamma f_l)$. The edges are oriented towards \mathcal{O}_F .

The gain of a closed (not directed) walk $e_1, e_2, e_3, \dots, e_k$ that starts at a vertex in \mathcal{O}_V is $\psi(e_1)\psi(e_2)^{-1}\psi(e_3)\dots\psi(e_k)^{-1}$. (Note that every other edge is used in the reverse direction; for these the inverse of their edge label is taken.) The *gain group* of a connected

edge set K and a vertex v spanned by K is defined by taking the set of gains of every closed walk in K starting with v . (Further investigations show that the choice of v can be arbitrary.) A connected edge set is *balanced*, if its gain group is the trivial group. Otherwise it is *unbalanced*. A not connected edge set is balanced, if it does not have an unbalanced component.

3.2 Necessary sparsity conditions for $d = 2$

Consider the special case when $d = 2$. Let $S(r)$ be a reflection-symmetric 1-picture. There are two choices for Γ' , namely \mathcal{C}_2 (half-turn) and \mathcal{C}_s (reflection). For these two symmetry groups we can give necessary conditions for the constraints to be independent.

Let (G_S, ψ) be the gain-bipartite graph of the incidence structure S . In order to determine independent constraints, every connected subgraph $G'_S = (V_1, F_1; E_1)$ of G_S has to satisfy the following two properties (for both \mathcal{C}_2 and \mathcal{C}_s):

1. for balanced sets $|E_1| \leq |V_1| + 2|F_1| - 2$;
2. for unbalanced sets we have $|E_1| \leq |V_1| + \sum_{f_j \in F_1} c_j - 1$ where $c_j = 1$ if $(v_i, f_j) \in I$ and $(\gamma(v_i), f_j) \in I$ for some i and $\gamma \neq \text{id}$ and $c_j = 2$ otherwise.

4 Further work

We expect that similar necessary conditions for forced symmetric liftings can also be established for higher dimensions. To obtain combinatorial characterisations, it is natural to consider inductive Henneberg-type construction moves. The results in [3] may also provide useful tools. These investigations are left for a future paper.

Acknowledgements

The second author was supported by the Hungarian Scientific Research Fund (OTKA, grant numbers FK128673, K124171).

References

- [1] **Kaszanitzky, V.E. and B. Schulze**, Lifting symmetric pictures to polyhedral scenes, *Ars Mathematica Contemporanea* **13** (1), 31-47
- [2] **Kaszanitzky, V.E. and B. Schulze**, Characterizing minimally flat symmetric hypergraphs, *Discrete Applied Mathematics* **236**, 256-269
- [3] **Tanigawa, S.**, Matroids of gain graphs in applied discrete geometry, *Trans. Amer. Math. Soc.* **367** (2015), 8597-8641
- [4] **Whiteley, W.**, A Matroid on Hypergraphs, with Applications in Scene Analysis and Geometry, *Discrete & Comput. Geom.* **4** (1989), 75-95
- [5] **Whiteley, W.**, Some Matroids from Discrete Applied Geometry, *Contemporary Mathematics, AMS* **197** (1996), 171-311

Algebraic matroids and global rigidity

Dániel Garamvölgyi

Department of Operations Research, Eötvös University, and the MTA-ELTE Egerváry
 Research Group on Combinatorial Optimization, Pázmány Péter sétány 1/C, 1117
 Budapest, Hungary.

daniel.garamvolgyi@ttk.elte.hu

Abstract

We give a characterization of globally rigid graphs in \mathbb{R}^d in terms of the natural algebraic representation of the d -dimensional rigidity matroid.

1 Definitions and main result

We start by recalling some definitions. In the following, let \mathbb{K} denote either \mathbb{R} or \mathbb{C} and fix $d \geq 1$. A *framework* in \mathbb{K}^d is a pair (G, p) where $G = (V, E)$ is a graph and p is a map from the vertex set V to \mathbb{K}^d . We also say that (G, p) is a *realization* of G in \mathbb{K}^d . The framework is *generic* if the set of its $|V| \cdot d$ coordinates is algebraically independent over \mathbb{Q} . Let n denote the number of vertices of G . For a pair of vertices $u, v \in V$, we define the map $m_{uv} : \mathbb{C}^{nd} \rightarrow \mathbb{C}$ by

$$m_{uv}(p) = \sum_{i=1}^d (p(u)_i - p(v)_i)^2,$$

where we are regarding the elements of \mathbb{C}^{nd} as maps from V to \mathbb{C}^d . The *d -dimensional edge measurement map* of G is the map $m_{d,G} : \mathbb{C}^{nd} \rightarrow \mathbb{C}^E$ given by

$$m_{d,G}(p) = (m_{uv}(p))_{uv \in E}.$$

We say that two frameworks (G, p) and (G, q) in \mathbb{K}^d are *equivalent* if $m_{d,G}(p) = m_{d,G}(q)$, and they are *congruent* if $m_{uv}(p) = m_{uv}(q)$ holds for all pairs of vertices $u, v \in V$. A framework (G, p) in \mathbb{K}^d is *rigid* if there is some $\varepsilon > 0$ such that every equivalent framework (G, q) in \mathbb{K}^d with $\|p - q\| < \varepsilon$ is, in fact, congruent to (G, p) . Here $\|\cdot\|$ denotes the Euclidean norm on \mathbb{K}^d . A framework (G, p) in \mathbb{K}^d is *globally rigid* if every equivalent framework in \mathbb{K}^d is congruent to it.

The graph G is *rigid in \mathbb{K}^d* (*globally rigid in \mathbb{K}^d* , respectively) if it has a generic rigid (globally rigid, resp.) realization in \mathbb{K}^d . This is equivalent to requiring that every generic realization in \mathbb{K}^d is rigid (globally rigid, resp.); see [1, 7] in the case of rigidity and [3, 6, 8] in the case of global rigidity. Furthermore, a graph is rigid in \mathbb{R}^d (globally rigid in \mathbb{R}^d , respectively) if and only if it is rigid in \mathbb{C}^d (globally rigid in \mathbb{C}^d , resp.), see [7, 8].

In the following, we shall assume that the reader is familiar with the basic notions of matroid theory. For a framework (G, p) in \mathbb{C}^d , let $R(G, p)$ denote the Jacobian of $m_{d,G}$ evaluated at p . Thus, $R(G, p)$ is a matrix with rows indexed by E and columns indexed by the nd coordinates of (G, p) . The rows of this matrix define a linear matroid on E .

It is folklore that this matroid is the same for every generic framework in \mathbb{C}^d ; this is the *d-dimensional generic rigidity matroid* of G which we denote by $\mathcal{R}_d(G)$.

Let R denote the polynomial ring over \mathbb{C} with indeterminates $x_v^i, v \in V, i = 1, \dots, d$ corresponding to the coordinate axes of \mathbb{C}^{nd} . By a slight abuse of notation, we shall also use m_{uv} to denote the polynomial

$$m_{uv} = \sum_{i=1}^d (x_u^i - x_v^i)^2 \in R.$$

The following result seems to be folklore. It follows e.g. from the Jacobi criterion of algebraic independence, see [4, Proposition 2.4].

Lemma 1 *The polynomials $m_{uv}, uv \in E$ give an algebraic representation of $\mathcal{R}_d(G)$ over \mathbb{R} . That is, a set $I \subseteq E$ is independent in $\mathcal{R}_d(G)$ if and only if $\{m_{uv} : uv \in I\}$ is algebraically independent over \mathbb{R} .*

Let K_V denote the complete graph on vertex set V . Our aim is to prove the following natural characterization of global rigidity in terms of the algebraic representation of $\mathcal{R}_d(K_V)$. Let $\text{Frac}(R)$ denote the field of fractions of the polynomial ring R .

Theorem 2 *A graph $G = (V, E)$ is globally rigid in \mathbb{R}^d if and only if the subfield of $\text{Frac}(R)$ generated by the polynomials $m_{uv}, uv \in E$ contains m_{uv} for every pair $u, v \in V$.*

Proof: This follows immediately from Theorem 5 below. □

The aim of the next section is to state and prove Theorem 5. In order to do this, we shall need to recall some basic results from algebraic geometry. For a detailed exposition, see e.g. [9]. See also [5] for more discussion on the measurement variety of graphs.

2 Global rigidity and algebraic geometry

For a set E , let $\mathbb{C}[E]$ ($\mathbb{Q}[E]$, respectively) denote the ring of polynomials over \mathbb{C} (over \mathbb{Q} , resp.) with indeterminates $x_e, e \in E$. An (*affine*) *variety* in \mathbb{C}^E is the set of simultaneous vanishing points of some polynomials $f_1, \dots, f_k \in \mathbb{C}[E]$. Affine varieties give the closed sets of the *Zariski topology* on \mathbb{C}^E .

A variety is *irreducible* if it cannot be written as the proper union of two varieties. A mapping $\varphi : X \rightarrow Y$ between varieties $X \subseteq \mathbb{C}^E$ and $Y \subseteq \mathbb{C}^{E'}$ is a *morphism* if the coordinate functions of φ are restrictions of polynomial functions $\mathbb{C}^E \rightarrow \mathbb{C}$ to X . The morphism is *dominant* if its image is Zariski-dense in Y . The *field of rational functions* $\mathbb{C}(X)$ of an irreducible variety $X \subseteq \mathbb{C}^E$ is the field of fractions of the quotient ring $\mathbb{C}[E]/I(X)$, where $I(X)$ is the set of polynomials vanishing on X . A dominant map $\varphi : X \rightarrow Y$ between irreducible varieties induces an inclusion of fields $\varphi^* : \mathbb{C}(Y) \rightarrow \mathbb{C}(X)$ via the (well-defined) mapping $f \mapsto f \circ \varphi$. The *degree* of φ is the degree of the field extension $\mathbb{C}(X) : \varphi^*(\mathbb{C}(Y))$. It is known that there is a Zariski-open subset of points $y \in Y$ such that the cardinality of the fiber $\varphi^{-1}(y)$ equals the degree of φ , see [9, Proposition 7.16]. If the degree of φ is one, we say that it is a *birational morphism*.

Now we specialize to the case of rigidity theory. Fix $d \geq 1$ and let $G = (V, E)$ be a graph. The *d-dimensional measurement variety* of G , denoted by $M_{d,G}$, is the Zariski-closure of

$m_{d,G}(\mathbb{C}^{nd})$, i.e. the smallest variety in \mathbb{C}^E that contains $m_{d,G}(\mathbb{C}^{nd})$. This is an irreducible variety. Recall the definition of the polynomial ring R above. The morphism $m_{d,G}$ induces an inclusion of fields $\mathbb{C}(M_{d,G}) \rightarrow \text{Frac}(R)$; in particular, $\mathbb{C}(M_{d,G})$ is isomorphic to the subfield of $\text{Frac}(R)$ generated by the polynomials $m_{uv}, uv \in E$.

For convenience, we shall use the following notion. Let (G, p) be a framework in \mathbb{C}^d and $u, v \in V$ a pair of vertices. We say that $\{u, v\}$ is *globally linked in (G, p)* if for every equivalent framework (G, q) in \mathbb{C}^d , we have $m_{uv}(p) = m_{uv}(q)$. The pair $\{u, v\}$ is *globally linked in G in \mathbb{C}^d* if it is globally linked in every generic realization of G in \mathbb{C}^d .

Finally, we shall need the following technical lemma, which can be proved by considering a basis of L as a K -vector space. See [2, Lemma A.2] for a similar statement.

Lemma 3 *Let $K \subseteq L$ be fields of characteristic zero and let $f_1, \dots, f_k, g \in K[x_1, \dots, x_n]$ be polynomials with coefficients in K , where x_1, \dots, x_n are independent transcendentals over L . Suppose that g lies in the subfield $L(f_1, \dots, f_k)$ of $L(x_1, \dots, x_n)$. Then $g \in K(f_1, \dots, f_k)$.*

Theorem 4 *Let $d \geq 1$ and let $G = (V, E)$ be a graph and $u, v \in V$ a pair of vertices. The following are equivalent.*

- a) *There exists a generic framework (G, p) in \mathbb{C}^d such that $\{u, v\}$ is globally linked in \mathbb{C}^d .*
- b) *The pair $\{u, v\}$ is globally linked in G in \mathbb{C}^d .*
- c) *The projection $\pi : M_{d,G+uv} \rightarrow M_{d,G}$ is a birational morphism.*
- d) *There exists a pair of polynomials $f, g \in \mathbb{Q}[E]$ where $g \circ m_{d,G}$ is not the zero polynomial and such that*

$$m_{uv} = \frac{f \circ m_{d,G}}{g \circ m_{d,G}},$$

that is, $m_{uv} \cdot (g \circ m_{d,G})$ and $f \circ m_{d,G}$ are equal as elements of the polynomial ring R .

Proof: (sketch) The equivalence of a) and b) follows from a standard genericity argument using Chevalley's theorem (see [8, Remark 2] for a similar argument). Assuming b), note that since $m_{d,G}$ is continuous, the image of generic points in \mathbb{C}^{nd} under $m_{d,G}$ forms a dense subset of $M_{d,G}$. Thus the assumption that $\{u, v\}$ is globally linked implies that the fiber of π has size one for a dense set in $M_{d,G}$. It follows that π has degree one, so it is a birational morphism.

For c) \Rightarrow d), recall that a birational morphism between two irreducible varieties induces an isomorphism between the fields of rational functions of the varieties. In our case, $\mathbb{C}(M_{d,G})$ is isomorphic to the subfield of $\text{Frac}(R)$ generated by $m_{u'v'}, u'v' \in E$ and π^* is the inclusion of this subfield into the subfield generated by $m_{u'v'}, u'v' \in E$ and m_{uv} . The fact that this inclusion is an isomorphism means that there is a pair of polynomials $f, g \in \mathbb{C}[E]$ with complex coefficients that satisfy the conditions of d). Now applying Lemma 3 with $K = \mathbb{Q}$ and $L = \mathbb{C}$ gives the desired result.

Finally, the d) \Rightarrow b) implication follows from the fact that, since $g \circ m_{d,G}$ is a non-zero polynomial with rational coefficients, $g \circ m_{d,G}(p)$ is non-zero for any generic realization (G, p) in \mathbb{C}^d . Thus, $m_{uv}(p)$ is uniquely determined by $m_{d,G}(p)$ for any generic realization (G, p) , as desired. \square

Theorem 5 *Let $d \geq 1$ and let $G = (V, E)$ be a graph on n vertices. The following are equivalent.*

- a) *G is globally rigid in \mathbb{R}^d .*
- b) *G is globally rigid in \mathbb{C}^d .*
- c) *The projection $\pi : M_{d, K^n} \rightarrow M_{d, G}$ is a birational morphism.*
- d) *For every pair of vertices $u, v \in V$ there exists a pair of polynomials $f_{uv}, g_{uv} \in \mathbb{Q}[E]$ where $g_{uv} \circ m_{d, G}$ is not the zero polynomial and such that*

$$m_{uv} = \frac{f_{uv} \circ m_{d, G}}{g_{uv} \circ m_{d, G}}.$$

Proof: The equivalence of a) and b) is [8, Theorem 1]. The rest of the equivalences follow immediately from Theorem 4 by noting that G is globally rigid in \mathbb{C}^d if and only if every pair of vertices $\{u, v\}$ is globally linked in G in \mathbb{C}^d . \square

3 Acknowledgements

This research was supported by the Hungarian Scientific Research Fund grant no. K135421, which has been implemented with the support provided from the National Research, Development and Innovation Fund of Hungary.

References

- [1] **L. Asimow and B. Roth**, The rigidity of graphs, *Trans. Amer. Math. Soc.*, **245** (1978), 279–289.
- [2] **M. Beecken, J. Mittmann and N. Saxena**, Algebraic independence and blackbox identity testing, *Information and Computation*, **222** (2013), 2–19.
- [3] **R. Connelly**, Generic global rigidity, *Discrete & Computational Geometry*, **33** (2005), 549–563.
- [4] **R. Ehrenborg and G. Rota**, Apolarity and Canonical Forms for Homogenous Polynomials *Europ. J. Combinatorics*, **14** (1993), 157–181.
- [5] **D. Garamvölgyi and T. Jordán**, Graph Reconstruction from Unlabeled Edge Lengths, *Discrete & Computational Geometry*, **66** (2021), 344–385.
- [6] **S. J. Gortler, A.D. Healy and D.P. Thurston**, Characterizing generic global rigidity, *American Journal of Mathematics*, **132** (2010), 897–939.
- [7] **S.J. Gortler, L. Theran and D.P. Thurston**, Generic unlabeled global rigidity, *Forum of Mathematics, Sigma*, **7** (2019), e21.
- [8] **S.J. Gortler and D.P. Thurston**, Generic global rigidity in complex and pseudo-Euclidean spaces. In: *Rigidity and Symmetry*, Fields Institute Communications **70**, Springer, New York (2014), 131–154.
- [9] **J. Harris**, *Algebraic geometry*, Springer-Verlag, New York, 1992.

Rigid block and hole graphs with a single block

Tibor Jordán

Department of Operations Research, ELTE Eötvös Loránd University, and the
MTA-ELTE Egerváry Research Group on Combinatorial Optimization, Eötvös Loránd
Research Network (ELKH), Pázmány Péter sétány 1/C, 1117 Budapest, Hungary

tibor.jordan@ttk.elte.hu

Abstract

Cruickshank, Kitson, and Power characterized the minimally rigid block and hole graphs G in \mathbb{R}^3 with a single block. This result, and the corresponding algorithm, can be used to verify rigidity only if G has exactly $3|V(G)| - 6$ edges. In this note we show that the constraint on the edge number can be omitted by providing an efficient algorithm which can determine whether a block and hole graph with a single block is rigid in \mathbb{R}^3 .

1 Introduction

A well-known result of rigidity theory¹, due to Gluck [4], states that every maximal planar graph (or *triangulation*) is generically rigid in \mathbb{R}^3 . Whiteley [8] initiated the study of the rigidity properties of modified triangulations (called *block and hole graphs*) that may contain blocks and holes. Extending the results of Finbow-Singh and Whiteley [3] on the single block and single hole case, Cruickshank, Kitson, and Power [2] characterized the minimally rigid block and hole graphs with a single block (and an arbitrary number of holes).

To describe their result we need the following definitions. Consider a planar embedding of a triangulation $G = (V, E)$. Note that G is 3-connected and the embedding is essentially unique. A cycle C of G divides the plane into two parts and hence it determines two subgraphs of G that share the edges and vertices of C . Such a subgraph is called a *disc*. We say that it is *bounded* by C , or that C is its *boundary cycle*. The *interior* of a disc consists of the vertices and edges of the disc that do not belong to its boundary cycle. Two discs are *internally disjoint* if their common edges or vertices, if they exist, are part of their boundary cycles.

We say that a *face graph* G^f of G is obtained from (a planar embedding of) G by choosing a collection of pairwise internally disjoint discs, removing the interiors of these discs, and then labeling the non-triangular faces of the resulting (embedded) planar graph by either b (block) or h (hole)². We may restrict ourselves to discs bounded by cycles of length at least four in G .

A *block-and-hole graph* G^\diamond with face graph G^f is obtained from G^f by adding new vertices and edges that rigidify the vertex set of each block. This is achieved as follows.

¹The reader is referred to [7] for an introduction to rigidity theory and further references.

²The definition of face graphs in [2, Definition 3] is slightly different. However, the authors (implicitly) use the definition given here.

Let C be the boundary cycle of a block-labeled face. We add two new vertices x_C and y_C as well as edges that connect these new vertices to each vertex of C . Then the vertex set $V(C) \cup \{x_C, y_C\}$ induces a bipyramid, which is a minimally rigid graph (a triangulation). We denote this subgraph by B_C . The block and hole graph is the union of the face graph and these bipyramids, one for each block-labeled face³. See Figure 1.

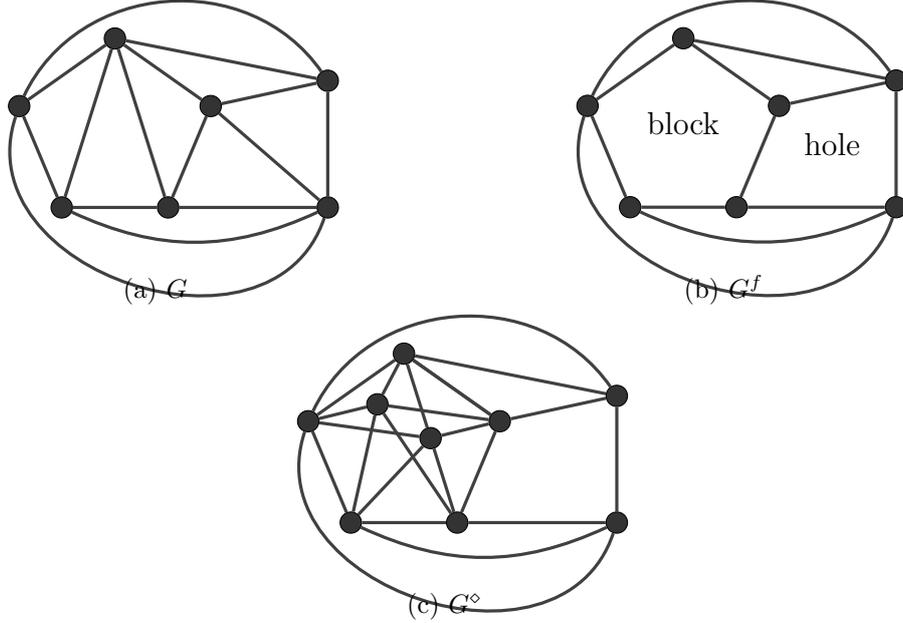


Figure 1: A triangulation G , a face graph G^f defined by two cycles of length five and four, respectively, and the block and hole graph G^\diamond . Graph G^\diamond is rigid, but not minimally rigid.

We say that a graph $G = (V, E)$ is $(3, 6)$ -sparse if $i_G(X) \leq 3|X| - 6$ for all $X \subseteq V$ with $|X| \geq 3$. Here $i_G(X)$ denotes the number of edges in the subgraph of G induced by the vertex set X . If G is a $(3, 6)$ -sparse graph with $|E| = 3|V| - 6$ then G is called $(3, 6)$ -tight. Note that in a simple graph no subset $X \subseteq V$ with $|X| \leq 4$ can violate the sparsity count. Moreover, the subsets $X \subseteq V$ with $|X| = 2$ satisfy the weaker bound $i(X) \leq 3|X| - 5$.

It is known that minimally rigid graphs in \mathbb{R}^3 are $(3, 6)$ -tight, but there exist non-rigid $(3, 6)$ -tight graphs. The main result of [2] shows that for a special family of block and hole graphs these two notions coincide.

Theorem 1 [2, Theorem 36] *Let G^\diamond be a block and hole graph with a single block. Then G^\diamond is minimally rigid in \mathbb{R}^3 if and only if G^\diamond is $(3, 6)$ -tight.*

In the next section we extend this result to (not necessarily minimally) rigid block and hole graphs.

³There are other ways to rigidify the block-labeled faces. Here we use this construction, which is called the *discus and hole graph* in [2].

2 Rigid block and hole graphs with a single block

We start with three simple observations. Recall that a graph $G = (V, E)$ on at least three vertices is *2-connected* if $G - v$ is connected for all $v \in V$.

Lemma 2 *Every face graph K is 2-connected.*

The statement of the lemma can be reversed in the following sense.

Lemma 3 *Let K be a 2-connected planar graph and let J be a non-triangular face in some planar embedding of K . Then there is a triangulation G for which K is a face graph of G in which J is a face.*

Lemma 4 *Suppose that $H = (V, F)$ is a maximal (with respect to edge inclusion) $(3, 6)$ -sparse subgraph of a 2-connected graph $G = (V, E)$. Then H is 2-connected.*

2.1 The characterization and the algorithm

Cheng and Sitharam [1] proved that the size of any maximal $(3, 6)$ -sparse subgraph of a graph G provides an upper bound for the rank of G in the 3-dimensional rigidity matroid. See also [5] for a different proof and extensions. Here we need the following corollary.

Theorem 5 [1] *Suppose that $G = (V, E)$ has a maximal $(3, 6)$ -sparse subgraph with less than $3|V| - 6$ edges. Then G is not rigid in \mathbb{R}^3 .*

We are ready to prove (the algorithmic version of) our main result.

Theorem 6 *Let G^\diamond be block and hole graph with a single block B^\diamond and let $H = (V(G^\diamond), F)$ be a maximal $(3, 6)$ -sparse subgraph of G^\diamond with $E(B^\diamond) \subseteq F$. Then G^\diamond is rigid if and only if H is $(3, 6)$ -tight.*

Proof: Since B^\diamond is minimally rigid, it is $(3, 6)$ -sparse. Thus we can extend (the edge set of) the block to a maximal $(3, 6)$ -sparse subgraph. Therefore a maximal $(3, 6)$ -sparse subgraph $H = (V(G^\diamond), F)$ with $E(B^\diamond) \subseteq F$ indeed exists.

Necessity follows from Theorem 5. To prove sufficiency we first use Lemmas 2 and 4 (and the fact that B^\diamond is 3-connected) to deduce that H is 2-connected. Let x and y be the new vertices (i.e. the poles) of the bipyramid B^\diamond . It is easy to see that $H - \{x, y\}$ is also 2-connected, and (as it is a subgraph of a face graph) has a planar embedding in which the cycle defining the block B^\diamond is the boundary cycle of some face J .

It follows from Lemma 3 that H is a block and hole graph with a single block (where the block labeled face in the corresponding face graph is J). We can now apply Theorem 1 to the $(3, 6)$ -tight graph H to deduce that it is (minimally) rigid. Since H is a spanning subgraph of G^\diamond , it follows that G^\diamond is also rigid, as required. QED.

We obtain the following characterization as a corollary.

Theorem 7 *Let G^\diamond be a block and hole graph with a single block. Then G^\diamond is rigid if and only if it has a minimally rigid spanning subgraph which is a block and hole graph with the same block.*

The results above imply that we can decide in polynomial time whether a block and hole graph G° with a single block is rigid. It relies on a subroutine which can test whether a given graph is $(3, 6)$ -sparse. See e.g. [6] for the description of such a subroutine which is based on matroidal methods and runs in polynomial time. With this subroutine in hand we can construct a maximal $(3, 6)$ -sparse subgraph H of G° , starting from B° , in a greedy manner.

3 Concluding remarks

It is also possible - by using similar methods - to compute the degree of freedom (i.e. the rank of G in the three-dimensional rigidity matroid) of a block and hole graph G with a single block in polynomial time. The algorithm for this more general problem will be given in the full version of this paper.

4 Acknowledgements

This work was supported by the Hungarian Scientific Research Fund grant no. K135421 and the project Application Domain Specific Highly Reliable IT Solutions which has been implemented with the support provided from the National Research, Development and Innovation Fund of Hungary, financed under the Thematic Excellence Programme TKP2020-NKA-06 (National Challenges Subprogramme) funding scheme.

References

- [1] **J. Cheng, M. Sitharam**, Maxwell-independence: a new rank estimate for the 3-dimensional generic rigidity matroid, *J. Combin. Theory, Ser. B*, 105, 2014, pp. 26-43.
- [2] **J. Cruickshank, D. Kitson, and S.C. Power**, The generic rigidity of triangulated spheres with blocks and holes, *J. Comb. Theory Ser. B.*, 122 (2017) 550-577.
- [3] **W. Finbow-Singh and W. Whiteley**, Isostatic block and hole frameworks, *SIAM J. Discrete Math.* 27 (2013) 991-1020.
- [4] **H. Gluck**, Almost all simply connected closed surfaces are rigid, in Geometric topology, Lecture Notes in Math., Vol. 438, Springer, Berlin, 1975, 225-239.
- [5] **H. Güler and B. Jackson**, A necessary condition for generic rigidity of bar-and-joint frameworks in d -space, *J. Graph Theory*, to appear, and arXiv:1104.4415, 2020.
- [6] **B. Jackson and T. Jordán**, On the rank function of the 3-dimensional rigidity matroid, *Int. J. Comp. Geom. and Applications*, Vol. 16, Nos. 5-6 (2006) 415-429.
- [7] **B. Schulze and W. Whiteley**, in: Rigidity and scene analysis, Handbook of Discrete and Computational Geometry, 3rd edition, J.E. Goodman, J. O'Rourke, C.D. Tóth eds, CRC Press, 2018.
- [8] **W. Whiteley**, Infinitesimally rigid polyhedra II: Modified spherical frameworks, *Trans. Amer. Math. Soc.* 306 (1988) 115-139.

Localizable sensor networks with optimal anchor sets I: A min-max theorem

Csaba Király^{1,2} and András Mihálykó¹

¹Department of Operations Research, ELTE Eötvös Loránd University

²MTA-ELTE Egerváry Research Group, Eötvös Loránd Research Network (ELKH)

{cskiraly,mihalyko}@cs.elte.hu

The two main concepts of rigidity theory are rigidity, where the framework has no continuous deformation, and global rigidity where the given distance set determines the locations of the points up to isometry. It is NP-hard to decide the rigidity or the global rigidity of a framework, however, in certain cases (for example, for *generic* frameworks in the plane where the coordinates of the points form an algebraically independent set over \mathbb{Q}), both the rigidity and the global rigidity of a framework depends only on the underlying graph. We consider the following problem. Given a set of sensors in the plane with known distances between some of them, at least how many sensor-locations do we need to measure exactly to be able to reconstruct the exact location of each sensor? This is the so-called *global rigidity pinning* (or anchoring) problem. To make this problem tractable, we assume that the locations of the sensors form a generic set.

Pinning can be modeled as adding a complete graph on the set of anchored vertices to the graph of known distances on the set of sensors as we can calculate the distances between any two pinned sensors exactly by their measured locations. It is known that, in the generic case in \mathbb{R}^2 , these distances uniquely determine the locations of the sensors if and only if the underlying graph is globally rigid (and there are at least 3 anchored sensors). This implies that the problem is equivalent to the following. Given a graph $G = (V, E)$, find a minimum size set $P \subseteq V$ such that $G \cup K_P$ is a globally rigid graph in the plane where K_P denotes the complete graph on the vertex set P . This problem was considered previously, and a constant factor approximation was given [1]. In this extended abstract, we give a min-max theorem yielding an optimal solution for a this problem provided that the input graph G is rigid. In Part II [5] we show how an $O(|V|^2)$ algorithm can be given to find this optimal vertex set. Moreover, we give a 2-approximation in case of non-rigid graphs, which improves the previous results for this problem.

1 Preliminaries

Rigidity of generic frameworks in \mathbb{R}^2 can be characterized by some sparsity properties of the underlying graph. A graph $G = (V, E)$ is called sparse, if $i_G(X) \leq 2|X| - 3$ for every $X \subseteq V$ where $|X| \geq 2$ and $i_G(X)$ denotes the number of edges of G induced by the set X . A sparse graph is tight, if $|E| = 2|V| - 3$. Following the famous theorems of Polaczek-Geiringer [8] and Laman [7] on the rigidity of generic frameworks in \mathbb{R}^2 , a graph is called rigid (in \mathbb{R}^2) if it contains a spanning tight subgraph.

The edge sets of the sparse subgraphs of a graph $G = (V, E)$ correspond to the independent sets of the so-called $(2, 3)$ -*sparsity matroid* (or *count matroid*) of G (see for example [2, Section 13.5]). The spanning tight subgraphs form a basis of this matroid. An edge

set which forms a circuit in this matroid, is called an *M-circuit*. It is well-known, that an equivalence relation can be defined on the ground set of an arbitrary matroid by using the circuit axioms. Two elements $e, f \in E$ of the $(2, 3)$ -sparsity matroid are equivalent if there exists an M-circuit C such that $e, f \in C$. The equivalence classes of this matroid are called *M-components* of G . The graph G is called *M-connected*, if it has only one M-component. The following theorem characterizes the globally rigid graphs in the plane.

Theorem 1 [3] *A graph $G = (V, E)$ with $|V| \geq 4$ is globally rigid in \mathbb{R}^2 if and only if G is 3-connected and M-connected.*

For a rigid graph $G = (V, E)$, let $\mathcal{H}_G = (V, \mathcal{E})$ be a hypergraph, called the *M-component hypergraph* of G , such that \mathcal{E} consists of $2|V(C)| - 3$ parallel copies of the hyperedge formed on $V(C)$ for each M-component C of G . Note that if an M-component consists of just one edge, \mathcal{H}_G contains the same edge. The definitions of sparse, tight and rigid graphs can be generalized to hypergraphs, in particular to the M-component hypergraphs, in a natural way.

Claim 2 *If G is a rigid graph on at least 4 vertices its M-component hypergraph \mathcal{H} is a tight hypergraph.*

A hypergraph (or a graph) \mathcal{H} is called *redundantly rigid*, if for any hyperedge (or edge, respectively) e of \mathcal{H} , the hypergraph (or graph, respectively) $\mathcal{H} - e$ is rigid.

Theorem 3 *Let $G = (V, E)$ be a rigid graph on at least 4 vertices and $\mathcal{H}_G = (V, \mathcal{E})$ be its M-component hypergraph. For a pinning set P , suppose that $G \cup K_P$ is 3-connected. Then the following statements are equivalent:*

- a) $G \cup K_P$ is redundantly rigid
- b) $\mathcal{H}_G \cup K_P$ is redundantly rigid
- c) $G \cup K_P$ is M-connected

By this theorem we aim to pin down G to 3-connected and \mathcal{H}_G to redundantly rigid. The advantage of this method is that we can use results on the redundant pinning structures of tight hypergraphs.

These results use the concept of *co-tight* sets. A vertex set C is a co-tight set of the hypergraph $\mathcal{H} = (V, \mathcal{E})$, if $|V - C| \geq 2$ and $V - C$ spans a tight hypergraph.

Theorem 4 [6] *Let $\mathcal{H} = (V, \mathcal{E})$ be a tight hypergraph on at least 4 vertices. Then $\min\{|P| : G \cup K_P \text{ is redundantly rigid}\} = \max\{|\mathcal{C}| : \mathcal{C} \text{ is a family of disjoint co-tight sets of } \mathcal{H}\}$. Moreover, P must intersect every co-tight set.*

It is easy to see that we might always pin down inclusion-wise minimal co-tight sets, that we denote with MCT sets for the sake of brevity. It is also known that the MCT sets of a tight hypergraph are often disjoint.

Lemma 5 [6] *Let \mathcal{H} be a tight hypergraph on at least 4 vertices. If there are at least 3 MCT sets of \mathcal{H} , then all the MCT sets are pairwise disjoint.*

We also need to pin down G to 3-connected thus we must consider its connectivity. First, it is a well-known folklore result, that every rigid graph is 2-connected. On the other hand, if there is a cut-pair $\{u, v\}$ of G , then every component of $G - \{u, v\}$ must be pinned down. A set P is called a *3-fragment* of a rigid graph G if $N_G(P)$ (the neighbor set of P in G) is a cut-pair in G and P induces a connected subgraph of G . Let us call the inclusion-wise minimal 3-fragments *3-ends*. The 3-ends of a rigid graph are pairwise disjoint [3]. It is easy to see, that each 3-end must be pinned down and that pinning down one vertex from each 3-end eliminates every cut-pair hence pinning down G to 3-connected.

2 The min-max theorem

As we saw, any pinning set must intersect all MCT sets and all 3-ends. Now we show how can we pin down G to globally rigid optimally.

Theorem 6 *Let $G = (V, E)$ be a rigid graph on at least 4 vertices. Let $\mathcal{H}_G = (V, \mathcal{E})$ be the M-component hypergraph of G . Then $\min\{|P| : G \cup K_P \text{ globally rigid}\} = \max\{|\mathcal{A}| : \mathcal{A} \text{ is a family of disjoint MCT sets of } \mathcal{H}_G \text{ and 3-ends of } G\}$.*

We sketch the main steps of the proof. First, if G is 3-connected, then Theorem 6 follows immediately by Theorem 4. Thus we may suppose that G is not 3-connected. Let us construct a set system from all the MCT sets of \mathcal{H}_G and 3-ends of G . Let \mathcal{A} contain the inclusion-wise minimal sets of this system. The sets contained in \mathcal{A} are called the *atoms* of G .

Lemma 7 *Let G be a rigid graph which is not 3-connected and let \mathcal{A} be the family of atoms of G . If $A, B \in \mathcal{A}$ then A and B are disjoint.*

A set P is called a *transversal* of \mathcal{A} if $|P \cap A| = 1$ for each $A \in \mathcal{A}$ and $|P| = |\mathcal{A}|$. By Lemma 7 it is easy to find a transversal of the atoms of G as one arbitrary vertex from each atom forms a transversal of \mathcal{A} .

Lemma 8 *Let G be a rigid graph which is not 3-connected and let \mathcal{A} be the atoms of G . If P is a transversal of \mathcal{A} , then $G \cup K_P$ is a globally rigid graph.*

We saw that one vertex in each atom must be pinned down and also that pinning down a transversal of the atoms results a globally rigid graph thus showing the optimality of pinning down a transversal of \mathcal{A} .

3 Non-rigid graphs

Theorem 6 leaves open the natural question, what can we do if G is not rigid. When G is not rigid, we can first pin G down to a rigid graph (which can be done optimally in polynomial time [1, 4]) and next pin this (already rigid graph) down to a globally rigid one.

If an optimal set that pins G down to a rigid graph is P_1 and an optimal set that pins $G \cup K_{P_1}$ down to a globally rigid graph is P_2 , while a minimal cardinality set pinning G down to globally rigid is P , it is easy to see that $|P_1| \leq |P|$ and $|P_2| \leq |P|$ hold hence $P_1 \cup P_2$ results a 2-approximation for the optimal pinning set in case of non-rigid inputs. This improves the previous result of 3-approximation [1].

Acknowledgements

Projects no. NKFI-128673 and K-135421 have been implemented with the support provided from the National Research, Development and Innovation Fund of Hungary, financed under the Hungarian Scientific Research Fund FK_18 and K_20 funding schemes. The first author was supported by the János Bolyai Research Scholarship of the Hungarian Academy of Sciences and by the ÚNKP-19-4 and ÚNKP-20-5. New National Excellence Program of the Ministry for Innovation and Technology. The second author was supported by the European Union, co-financed by the European Social Fund (EFOP-3.6.3-VEKOP-16-2017-00002). The authors are grateful to Tibor Jordán for the inspiring discussions and his comments.

References

- [1] **Fekete, Zs. and Jordán, T.**, Uniquely localizable networks with few anchors. In S.E. Nikolettseas and J.D.P. Rolim (Eds.), *Algorithmic Aspects of Wireless Sensor Networks*, Springer Berlin Heidelberg (2006), 176–183.
- [2] **Frank, A.**, Connections in Combinatorial Optimization. *Oxford University Press* (2011)
- [3] **Jackson, B. and Jordán, T.**, Connected rigidity matroids and unique realizations of graphs, *J. Combin. Theory, Ser. B*, **94** (2005), 1–29.
- [4] **Jordán, T.**, Combinatorial rigidity: Graphs and matroids in the theory of rigid frameworks. In *Discrete Geometric Analysis, MSJ Memoirs*, **34** (2016), 33–112.
- [5] **Király, Cs. and Mihálykó, A.**, Localizable sensor networks with optimal anchor sets I: A min-max theorem, in *Proceedings of Developments in Computer Science*, ELTE, Hungary (2021).
- [6] **Király, Cs. and Mihálykó, A.**, Sparse graphs and an augmentation problem. Technical Report TR-2020-06, Egerváry Research Group, Budapest (2020). www.cs.elte.hu/egres
- [7] **Laman, G.**, On graphs and rigidity of plane skeletal structures. *J. Engineering Mathematics*, **4** (1970), 331–340.
- [8] **Pollaczek-Geiringer, H.**, Über die Gliederung ebener Fachwerke. *ZAMM - Journal of Applied Mathematics and Mechanics*, **7** (1927), 58–72.

Localizable sensor networks with optimal anchor sets II: An algorithm

Csaba Király^{1,2} and András Mihálykó¹

¹Department of Operations Research, ELTE Eötvös Loránd University

²MTA-ELTE Egerváry Research Group, Eötvös Loránd Research Network (ELKH)

{cskiraly,mihalyko}@cs.elte.hu

In this extended abstract we show how one can provide an efficient algorithm for the *global rigidity pinning problem*. We show that, if the input graph $G = (V, E)$ is *rigid*, then there exists an algorithm with running time $O(|V|^2)$ which finds a minimum subset $P \subseteq V$ for which $G \cup K_P$ is *globally rigid* (that is, *3-connected* and *redundantly rigid* by [5]), where K_P is the complete graph on P .

We shall use the notions and theorems from Part I of this sequence of papers [6]. As described in [6], a minimum pinning set can usually be given by providing a transversal of the family of the *atoms* of G which are the minimal elements of the union of the family of the *MCT sets* of the *M-component hypergraph* \mathcal{H}_G of G and of the family of the *3-ends* of G . (The only exception is in the case where G is already 3-connected and we only need to augment \mathcal{H}_G to redundantly rigid, however, in this case earlier methods from [7] can be applied.) To give an $O(|V|^2)$ time algorithm, we first need to construct the M-component hypergraph \mathcal{H}_G in this running time. The 3-ends of G can be found by the linear time 3-connectivity augmentation algorithm of Hsu [4]. This algorithm has $O(|V| + |E|)$ running time. Finally, by slightly modifying an algorithm by García and Tejel [2], we can output an optimum pinning set.

1 The algorithmic construction of the M-component hypergraph

Let us first briefly summarize the algorithm for testing rigidity and its main properties (see [1, 8] for more details). This algorithm is based on the Orientation Lemma of Hakimi [3] and uses in-degree constrained orientations. To check the rigidity of G , the algorithm constructs a tight subgraph of G by considering its edges one by one. During the construction we have a sparse subgraph G' of G and we also maintain an orientation \vec{G}' of G' in which the in-degree of each vertex is at most two. Before adding an edge ij to G' , we try to find a reorientation of G' in which the in-degree of i and j is zero (while all the other in-degrees are at most two). Such reorientation (if exists) can be found in $O(|V|)$ time by running constant number of backward DFS on \vec{G}' from i and j (and switching the orientation of paths ending at i or j). If we find a proper reorientation, then, by the Orientation Lemma [3], each set X containing both i and j induces at most $2|X| - 4$ edges in G' and hence adding the edge ij to G' maintains its sparsity. Otherwise, the edge ij can be omitted, moreover, (during the backward DFS) we find a minimum set containing both i and j with in-degree zero in \vec{G}' . This is the minimum set X which contains i and j and induces $2|X| - 3$ edges in G' , that is, the vertex set of the minimum tight subgraph of G' containing both i and j .

Lee and Streinu [8] (by using a data structure discovered in their joint paper with Theran [9]) reduced the total running time of this algorithm from $O(|V||E|)$ to $O(|V|^2)$. The main idea is to maintain the family of rigid components of G' which structure can be used to omit edges, which are not needed to construct a tight subgraph, in constant time. It is easy to update the structure of rigid components in $O(|V|)$ time when we add an edge. Beside this update, we need a data structure which helps to decide in constant time whether two vertices are in the same component. To get an easily updatable data structure which fulfills this goal [9], we consider the edges in a *breath-first* manner, that is, in such a way that we take all the edges incident with the same vertex $v \in V$ in a row, and, as long as we consider these edges, we maintain a 0-1 vector of length $|V| - 1$ which is one at a coordinate corresponding to a vertex $w \in V - v$ if and only if v and w are induced by a tight subgraph (that is, a rigid component) of G' . When we add a new edge to G' or start to consider the edges incident with another vertex, this vector can be updated in $O(|V|)$ time. This along with the constant time omission of unusable edges implies a total running time of $O(|V|^2)$ for the rigidity testing algorithm.

The above algorithm also works for testing the sparsity and tightness of a hypergraph [10], however, its running time is slightly worse in the general case since the size of the hyperedges affect the running time of the backward DFS subroutines. Fortunately, the M-component hypergraph has $2|e| - 3$ parallel copies of each hyperedge e which implies that the running time in this special case will not be increased.

We shall use the above idea for the construction of the M-component hypergraph in $O(|V|^2)$ time with the following differences. First, instead of maintaining G' , \vec{G}' , the rigid components of G , and the above mentioned 0-1 vector, we maintain the M-component hypergraph $\mathcal{H}_{G'}$ of the already considered edges, its orientation $\vec{\mathcal{H}}_{G'}$ in which the in-degree of each vertex is at most two (and each hyperedge has one head and multiple tails), and a 0-1 vector c of length $|V| - 1$ which tells us whether the currently considered vertex is contained in the same M-component with other vertices (that is, this vector is one at the $\mathcal{H}_{G'}$ -neighbors of the currently considered vertex).

As before, we consider the edges one by one in a breath first manner. When we consider an edge ij , we first check in our vector whether i and j are contained in the same M-component. If yes, then the edge ij is useless (that is, it cannot be used to construct larger M-components) and hence we can omit it which only need constant time in this case. Otherwise, we try to find a reorientation of $\vec{\mathcal{H}}_{G'}$ in which the in-degree of i and j are zero and all the other in-degrees are at most two by performing constant number of backward DFSs from i and j (and switching the orientation of paths ending at i or j). Each of these DFSs needs $O(|V|)$ running time since the edge number in the sparse hypergraph $\mathcal{H}_{G'}$ is $O(|V|)$ and furthermore we only need to consider one of the $2|e| - 3$ parallel copies of the same hyperedge e during the search. If we can find a proper reorientation, then we add ij to $\mathcal{H}_{G'}$, orient it arbitrarily, and modify the vector c correspondingly. Note that we add $2|V| - 3 = O(|V|)$ edges this way to $\mathcal{H}_{G'}$ during the algorithm (which edges form a tight subgraph of G). If we cannot find a proper reorientation, we find a minimum set X which has in-degree zero in $\vec{\mathcal{H}}_{G'}$ and contains both of i and j , that is, the minimum set X which contains both of i and j and induces $2|X| - 3$ hyperedges in $\mathcal{H}_{G'}$. Then we update $\mathcal{H}_{G'}$ by deleting all hyperedges induced by X and adding $2|X| - 3$ copies of X to its hyperedge set. The heads of these hyperedges in $\vec{\mathcal{H}}_{G'}$ will be exactly the heads of the omitted hyperedges.

This update along with the update of the vector c needs $O(|V|)$ running time (see also [9]). Note that during such a step we merge at least two M-components and hence we have at most $2|V| - 3 = O(|V|)$ steps of this type. This implies that the total running time of the algorithm is $O(|V|^2)$. The following statement shows that this way we indeed get the M-component hypergraph.

Lemma 1 *Let G' be a graph and let $\mathcal{H}_{G'} = (V, \mathcal{E})$ be its M-component hypergraph. Let ij be an edge which is not induced by the vertex set of any M-component of G' . Assume that i and j are induced by a tight subhypergraph of $\mathcal{H}_{G'}$ and let X be the vertex set of the minimum tight such subhypergraph of $\mathcal{H}_{G'}$. Then the M-component hypergraph of $G' + ij$ arises from $\mathcal{H}_{G'}$ by deleting its subhypergraph $\mathcal{H}_{G'}[X]$ induced by X and adding $2|X| - 3$ copies of the hyperedge X .*

Proof: (Sketch.) Let G'' be a maximal sparse spanning subgraph of G' . It is easy to see that X induces a tight subgraph in G'' containing i and j . Hence the vertex set Y of the minimum such subgraph is a subset of X , and thus $G' + ij$ will not contain any M-circuit which is not a subgraph of $G''[X]$. On the other hand, we claim that X contains at least two vertices from the vertex set of each M-components of G' . To see this, observe that the vertex set of each M-component of G' which intersects Y by at least two vertices can be added to Y by maintaining the tightness of the induced subgraph. Moreover, the vertex set, which we get with this method, induces a tight subhypergraph of $\mathcal{H}_{G'}$ containing both i and j , hence it must be equal to X . This implies that each M-component of G' , which is induced by a subset of X , has at least one edge contained in an M-circuit of $G' + ij$ containing ij and hence the union of these M-components of G' (along with the edge ij) form an M-component in $G' + ij$. \square

2 The construction of a minimum pinning set

To construct a minimum set for which $G \cup K_P$ is globally rigid, we shall use the main idea of an algorithm by García and Tejel [2]. This algorithm takes a minimum degree vertex i of a tight graph G and runs a subroutine called *Find a minimum covering rooted at i* with this vertex. This greedy subroutine sequentially calculates, for each unmarked vertex v , the vertex set T_{iv} of the minimum tight subgraph of G containing i and v , marks the elements of T_{iv} and modifies its output V' to $(V' - X) + v$. [7, Lemma 5.10] and the results of [2] imply that V' or $V' + i$ is a transversal of the MCT sets of G . This can be proved by using the fact that $T_{iu} \cap T_{iv}$ induce at least one edge of G for each $u, v \in V'$ due to the minimum degree of i . To decide which one of these two possibilities is our case, we only need to run again *Find a minimum covering rooted at v* for a vertex $v \in V$.

Now we modify the above algorithm, as follows. We first take a tight spanning subgraph of G and search for a minimum degree vertex i in this graph. We shall run *Find a minimum covering rooted at i* in the tight hypergraph \mathcal{H}_G , and then, for a vertex v from the output, *Find a minimum covering rooted at v* again in \mathcal{H}_G . Note that either i has degree at most three in \mathcal{H}_G or it is incident with the parallel copies of one hyperedge and with at most one normal edge in \mathcal{H}_G . This implies that the previous proof method can be used to prove that the output V' of the second run of *Find a minimum covering* is a transversal of the MCT sets. Moreover, it follows from the proof that we put the first vertex which we

explore from an MCT set into V' , hence if we explore first the (unmarked) vertices from the 3-ends of G , then each MCT set which intersects a 3-end will be represented by a vertex from a 3-end. Hence to get a transversal of the atoms we only need to add a vertex to V' from each 3-end which is not intersected by V' . As we have seen earlier, the vertex set of a minimum tight subhypergraph of \mathcal{H}_G containing i and v can be calculated in $O(|V|)$ time (by using the 2-in-degree constrained orientation of \mathcal{H}_G). This implies that the total running time of our algorithm is $O(|V|^2)$.

Acknowledgments

Projects no. NKFI-128673 and K-135421 have been implemented with the support provided from the National Research, Development and Innovation Fund of Hungary, financed under the Hungarian Scientific Research Fund FK_18 and K_20 funding schemes. The first author was supported by the János Bolyai Research Scholarship of the Hungarian Academy of Sciences and by the ÚNKP-20-5 New National Excellence Program of the Ministry for Innovation and Technology. The second author was supported by the European Union, co-financed by the European Social Fund (EFOP-3.6.3-VEKOP-16-2017-00002). The authors are grateful to Tibor Jordán for the inspiring discussions and his comments.

References

- [1] **Berg, A. and Jordán, T.**, Algorithms for Graph Rigidity and Scene Analysis, in G. Di Battista and U. Zwick (Eds.) *Algorithms - ESA 2003* (Budapest, 2003), LNCS **2832**, Springer, 2003, 78–89.
- [2] **García, A. and Tejel, J.** Augmenting the Rigidity of a Graph in \mathbb{R}^2 , *Algorithmica*, **59**(2) (2011), 145–168.
- [3] **Hakimi, S.**, On the degrees of the vertices of a directed graph, *J. Franklin Inst.*, **279**(4) (1969), 290–308.
- [4] **Hsu, T. and Ramachandran, V.**, A linear time algorithm for triconnectivity augmentation, in *Annual Symposium on Foundations of Computer Science (Proceedings)*, 1991, 548–559.
- [5] **Jackson, B. and Jordán, T.**, Connected rigidity matroids and unique realizations of graphs, *J. Combin. Theory, Ser. B*, **94** (2005), 1–29.
- [6] **Király, Cs. and Mihálykó, A.**, Localizable sensor networks with optimal anchor sets I: A min-max theorem, in *Proceedings of Developments in Computer Science*, ELTE, Hungary, 2021.
- [7] **Király, Cs. and Mihálykó, A.**, Sparse graphs and an augmentation problem. Technical Report TR-2020-06, Egerváry Research Group, Budapest, 2020. www.cs.elte.hu/egres
- [8] **Lee, A. and Streinu, I.**, Pebble game algorithms and sparse graphs, *Discrete Math.*, **308**(8) (2008), 1425–37.
- [9] **Lee, A., Streinu, I., and Theran, L.**, Finding and Maintaining Rigid Components, in *Proceedings of the 17th Canadian Conference on Computational Geometry, CCCG'05*, (University of Windsor, Ontario, Canada, August 10–12, 2005), 2005, 219–22.
- [10] **Lee, A., Streinu, I., and Theran, L.**, Sparse hypergraphs and pebble game algorithms, *European J. Combin.*, **30**(8) (2009), 1944–64.

Section:

Information Systems and Architectures

Organizer: Bálint Molnár

Introductory talk:

Bálint Molnár: Formal approaches for modelling Information Systems

Contributions:

- Balázs Horváth and Bálint Molnár: Dynamic process modeling of micro-credentials
- Meriem Kherbouche, Ahmad Mukashaty and Bálint Molnár: An Operationalized Transformation for Activity Diagram into YAWL
- Zhang Yinghong and Bálint Molnár: An overview of reinforcement learning applications in the control system of the intelligent transportation system
- Ekaterina Zolotareva, Bethelihem Seifu and Bálint Molnár: Credit risk management in financial services



Formal approaches for modelling Information Systems

(Introductory talk)

Bálint Molnár  0000-0001-5015-8883

Information Systems Department, Eötvös Loránd University of Budapest, ELTE IK,
Pázmány Péter sétány 1/C, 1117 Budapest, Hungary

`molnarba@inf.elte.hu`

Abstract The document-centric perception of Information Systems became ubiquitous recently. Not only the output, input, and internally stored data collections appear as documents but the representation of Business Processes and the artifacts of Enterprise Architecture. For this reason, the document-centric approaches for understanding the behavior of Information systems are an apt tool. Information Systems can be considered from three viewpoints. Namely data, processes, and behavior that represents the complex interaction of business processes through the data. The representation of this complex set of relationships requires approaches grounded in various branches of Mathematics and Computer Science, namely Information Theory, Computational Graph Theory, and Computational Algebraic Topology. In this paper, the essential ideas, proposed methods, and approaches are outlined to open a way for further researches.

1 Introduction

The Architecture of Information Systems covers several aspects, perspectives, viewpoints that expound the relevant features of an operating or being under design Information System. Emerging standards in past decades like XML [5, 9] made it possible that various constituents of Information Systems can be described and represented by document formats. The interchange format of data collections is XML, JSON that can represent traditional Relational Databases but XML, JSON is the *lingua franca* for the most recent heterogeneous structure of various databases [8, 6, 10, 4, 16]. Thus, the input, output, content and Business Processes can be described by various documents having standard specification languages.[7, 14].

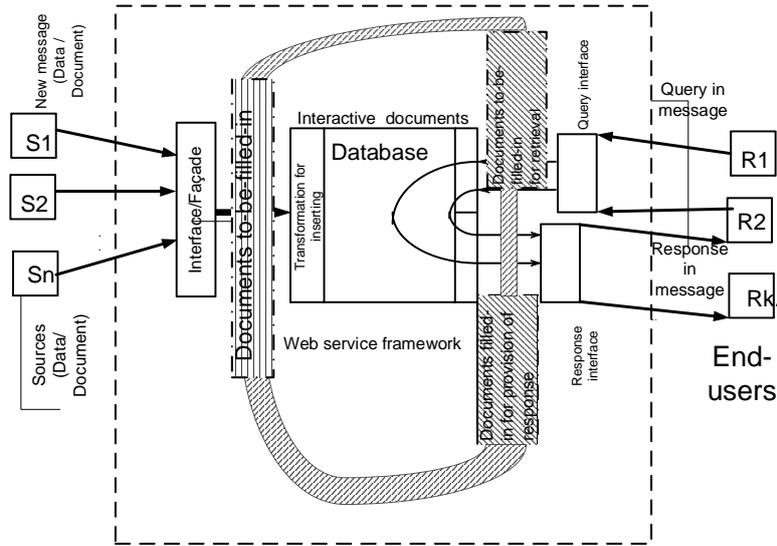


Figure 1: Information Systems' model in a document-centric and Information Theoretic viewpoint based on [13]

The various facets of Information Systems can be described by documents that can be represented by graph structures. The complex relationships within a facet and mutual interactions among them can be represented by hypergraphs. Since, the hypergraphs are able to depict multi-dimensional relationships among multi-sets. The hypergraph description of the models of Information Systems that can be placed into Enterprise Architecture frameworks as Zachman or Togaf [15, 17] can be analyzed with the help of mathematical tool sets.

2 Discussion, analysis

To exploit the mathematical toolbox, the first task is to represent models of the fundamental aspects of information systems. The first task was to find a formal approach to describing documents in a comprehensive way that embraces the generic documents and their step-by-step refinements till the finalized ones [13]. The hypergraph can be transformed into a bipartite graph that can be represented in graph databases that are accessible [8, 6, 10](see Figure 2). The second task was to customize the searching algorithm available for graphs for our purpose to find an effective and efficient set of algorithms that can be used to locate patterns and phenomena and lay the foundation for visualization. The tailoring and systemic investigation of the algorithms happened in an experimental design and development. Both the performance and complexity analysis were carried out along with the operationalization of algorithms in a graph database[12].

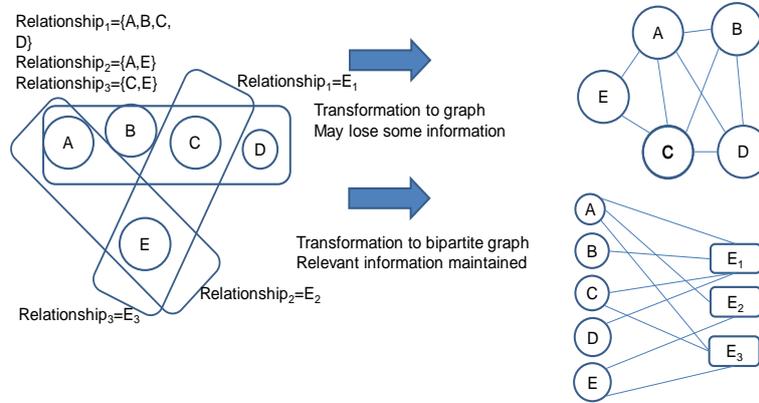


Figure 2: The transformation of a hypergraph that represents the relationships among data

Thus, the customized algorithm can be exploited to analyze the represented model. One of the models is the document model that can include the stored data and their representation by schemata for data collections. The other model is the model of Business Processes that describes the behavior of the Information System. The document- and data-centric models represented by hypergraphs are interconnected to the behavior model, to the representation of Business processes through complex relationships.

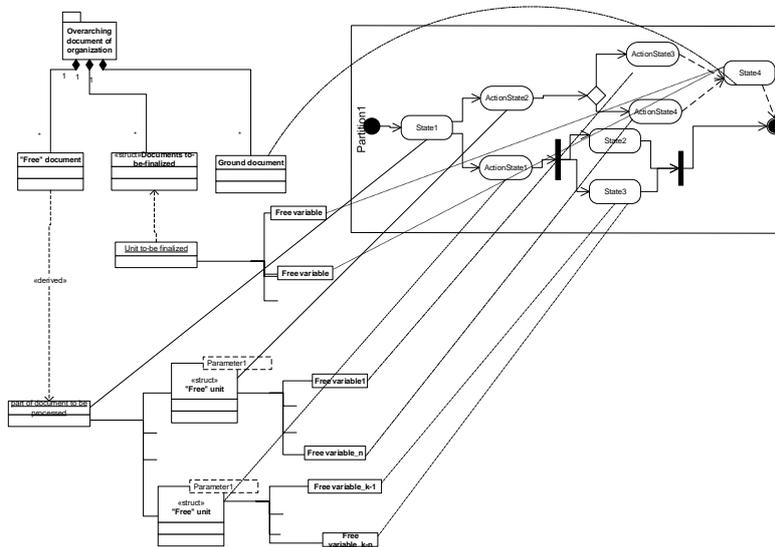


Figure 3: Interactions between a Business Process and the document structure

The major aim is to check the integrity, consistency, and certain security properties of the system [2] (see 3). The transformation of hypergraphs into bipartite graphs can be

applied for the representation of Business Processes. The graph representation including the hypergraphs offers the opportunity to utilize linear algebra and algebraic topology methods for the analysis of the underlying structure. One of the possibilities is to calculate the Smith Normal Form for single graphs and investigate the similarities and dissimilarities of the individual graphs. An experimental design and software experiment was carried out that used Smith Normal Form and clustering of represented Business Processes to discover significant problems in changes of Business Process in a dynamic environment [1]. Besides Smith Normal Form as a linear algebraic approach, the hypergraphs can be mapped onto simplicial complexes that yield the opportunity to use the mathematical toolset of algebraic topology. The feature of simplicial complexes, which represent hypergraphs, can be examined by the structures of homology groups and their invariants, e.g. Betti numbers [3]. The homology groups make it possible to identify holes within simplicial complexes, e.g. a 2-dimension hole within a simplicial complex has a 1-dimension boundary that is a loop, a 3-dimension hole is the void within a torus that has a boundary defines by a 2-dimension surface, thereby the "loops" of hypergraphs can be explored as well. To exploit the mathematical structures of a model representing hypergraphs, appropriate coding is required, i.e. the hypergraphs of models in both structure and explanatory abilities should be integrated to give understandable results. Besides the structural analysis, we can apply logic in the form of Description Logic and Horn function to carry out reasoning on the hypergraphs structures, an application of Description Logic is showcases in Reference [11].

3 Conclusion and Future Work

The Information Systems embody an interdisciplinary field between Computer Science, Informatics, and Management Sciences. Application of formal methods of Mathematics and Computer Science proffer the opportunity that the changes that are enforced by the environment of Information Systems can be kept in hand. The complexity of the situation can be handled by a Virtual Twin solution when a well-defined part of the Information System is replicated in a virtual environment that allows for the experimenting and studying the various phenomena of a productive Information System. The results that are achieved up to now seem promising, namely, the operationalization of searching algorithm on hypergraphs, the utilization of Smith Normal Form with the combination of Data Science algorithms. The theoretical foundation is the formalized description of Information Systems in hypergraphs is progressed to laying the groundwork for the description of various facets of Information Systems as documents, data, and Business Processes.

In future work, the research would aim at the exploitation of homology groups and Horn function to explore the different phenomena of the specific models in Information Systems. The model checking of Information Systems can use the progress of Computational Topology and the development of the relevant algorithms that can be accessed in open-source format. The formally defined single models of Information Systems can be embedded into a Virtual Twin environment where the models that are specified by hypergraphs then the graph representations are transformed into other mathematical structure where the mathematical methods and their operationalized algorithms can be employed for model checking and discovering anomalies, patterns, and phenomena.

Acknowledgment The project was supported by the European Union, co-financed

by the European Social Fund (EFOP-3.6.3-VEKOP-16-2017-00002) and the project was partially supported by „Application Domain Specific Highly Reliable IT Solutions” project that has been implemented with the support provided from the National Research, Development and Innovation Fund of Hungary, financed under the Thematic Excellence Programme TKP2020-NKA-06 (National Challenges Subprogramme) funding scheme.

References

- [1] Khawla Bouafia, Maxim Kumundzhiev, and Bálint Molnár. Application of models and hypergraph on dynamic aspect of business process performance analysis. *Annales Universitatis Scientiarum Budapestinensis de Rolando Eötvös Nominatae. Sectio Computatorica*, oct 2020. Submitted.
- [2] Edmund M. Clarke, Thomas A. Henzinger, Helmut Veith, and Roderick Bloem, editors. *Handbook of Model Checking*. Springer International Publishing, 2018. doi:10.1007/978-3-319-10575-8.
- [3] Jackson Earl. Computing homology of hypergraphs, 2019. arXiv:<https://digitalcommons.calpoly.edu/star/561>.
- [4] Bryon K. Ehlmann. *Object Relationship Notation (ORN) for Database Applications*. Springer US, 2009. doi:10.1007/978-0-387-09554-7.
- [5] Jeff Friesen. *Java XML and JSON*. Apress, 2019. doi:10.1007/978-1-4842-4330-5.
- [6] Borislav Iordanov. Hypergraphdb: A generalized graph database. In Heng Tao Shen, Jian Pei, M. Tamer Özsu, Lei Zou, Jiaheng Lu, Tok Wang Ling, Ge Yu, Yi Zhuang, and Jie Shao, editors, *Web-Age Information Management - WAIM 2010 International Workshops: IWGD 2010, XMLDM 2010, WCMT 2010, Jiuzhaigou Valley, China, July 15-17, 2010, Revised Selected Papers*, volume 6185 of *Lecture Notes in Computer Science*, pages 25–36. Springer, 2010. URL: https://doi.org/10.1007/978-3-642-16720-1_3, doi:10.1007/978-3-642-16720-1_3.
- [7] Hyoung Do Kim. Conceptual modeling and specification generation for b2b business processes based on ebXML. *ACM SIGMOD Record*, 31(1):37–42, mar 2002. doi:10.1145/507338.507346.
- [8] Kobrix Software. Hypergraphdb - a graph database, 2010. arXiv:<http://hypergraphdb.org>.
- [9] Joe Marini. *Document Object Model*. McGraw-Hill, Inc., 2002.
- [10] Rudolf Michael, Paradies Marcus, Bornhövd Christof, and Lehner Wolfgang. The graph story of the SAP HANA database. In Volker Markl, Gunter Saake, Kai-Uwe Sattler, Gregor Hackenbroich, Bernhard Mitschang, Theo Härder, and Veit Köppen, editors, *Datenbanksysteme für Business, Technologie und Web (BTW), 15. Fachtagung des GI-Fachbereichs "Datenbanken und Informationssysteme" (DBIS), 11.-15.3.2013 in Magdeburg, Germany. Proceedings*, volume P-214 of *LNI*, pages 403–420. GI, 2013. URL: <https://dl.gi.de/20.500.12116/17334>.

- [11] B. Molnár, A. Béleczi, and A. Benczúr. Information systems modelling based on graph-theoretic background. *Journal of Information and Telecommunication*, 2(1):68–90, sep 2017. doi:10.1080/24751839.2017.1375223.
- [12] Bálint Molnár, András Béleczi, and Bence Sarkadi-Nagy. Storing hypergraph-based data models in non-hypergraph data storage and applications for information systems. *Vietnam Journal of Computer Science*, 8(3), aug 2021.
- [13] Bálint Molnár and András Benczúr. Issues of modeling web information systems proposal for a document-centric approach. *Procedia Technology*, 9:340–350, 2013. doi:10.1016/j.protcy.2013.12.038.
- [14] Chul-Ki Nam, Gil-Sang Jang, and Jae-Hak J. Bae. An xml-based active document for intelligent web applications. *Expert Systems with Applications*, 25(2):165–176, 2003. doi:10.1016/s0957-4174(03)00044-7.
- [15] Open Group. Togaf: The open group architecture framework, version 9. electronic, 2010. URL: <http://www.opengroup.org/togaf/>, arXiv:<http://www.opengroup.org/togaf/>.
- [16] John Tomcy and Misra Pankaj. *Data Lake for Enterprises*. Packt Publishing, 2017. URL: https://www.ebook.de/de/product/29271474/tomcy_john_pankaj_misra_data_lake_for_enterprises.html.
- [17] John A. Zachman. A framework for information systems architecture. *IBM systems journal*, 26(3):276–292, 1987.

Dynamic process modeling of micro-credentials

Balázs Horváth  0000-0002-1772-3067 , Bálint Molnár  0000-0001-5015-8883

Information Systems Department, Eötvös Loránd University of Budapest, ELTE IK,
Pázmány Péter sétány 1/C, 1117 Budapest, Hungary
(hobuabi, molnarba) @inf.elte.hu

Abstract Education arrived to a new step, where modularity is playing an important role in lifelong learning. Young professionals continuously want to pick up new skills but not necessary going through a full university program. Micro-credentials offer a great solution to prove certain obtained skills. The role of higher education institutes is changing with these novel certification methods. Academic recognition is crucial to ensure acceptance and uptake of micro-credentials within higher education. In order to issue a diploma to the learners, they have to show they fulfill the rules of the institute, which can be checked using process mining techniques. Conformance checking runs process models against pre-set rules to check if that model complies with them or not. These techniques with the proper process modelling tools can provide the option to higher education institutes to automatically issue certificates even diplomas if the learner obtained the necessary micro-credentials to get them.

1 Introduction

Process mining is not an unknown domain in the education field, several research used it to enhance the quality of courses (mostly on MOOCs, due to the easy access to learning process data). An example is [2], where the authors show the advantages of process mining with process cubes on educational data.

This research identifies the best process modelling techniques for the purpose. Not only the micro-credential granting process can be enhanced with process mining but the evaluation of the learning process as well. However, a fair evaluation can only be carried out if transparent information is available on elements [1] such as the process, quality, workload, level and learning outcomes of the credential. In this research the focus lays on the learning process modelling. The conventional process modelling techniques are not suitable enough for making it transparent and optimal for conformance checking algorithms. These conventional modelling techniques (such as BPMN, Petri nets, Workflow nets, UML, etc...) on dynamic processes like the learning path of a MOOC results in a so called spaghetti model, which is hard to interpret and also graph algorithms perform poorly on them. Lastly the focus lays on a hypergraph based modeling technique (Flexible Process graph [6]), which offers a suitable solution to the micro-credential modeling task with few modifications.

2 Discussion, analysis

For the comparison of different modelling techniques a tailored taxonomy [3], which is based on the Zachman or Togaf [5, 8] frameworks, was used.

For ad-hoc process modeling purposes [6] created a hypergraph based solution called the Flexible process graph and its definition is the following:

Definition 1 *The Flexible process graph is a triplet (A, E, T) where:*

A is a finite set of activity nodes

E is a finite set of edges e , where $e = \langle I(e), O(e) \rangle \in E, A \cap E = \emptyset$

$I : E \rightarrow P(A)$ is a function defining edge input activities

$O : E \rightarrow P(A) \setminus \emptyset$ is a function defining edge output activities

$\forall e \in E : I(e) \cap O(e) = \emptyset$

T is an edge type function $T : E \rightarrow \{\mathcal{AND}, \mathcal{OR}, \mathcal{XOR}\}$

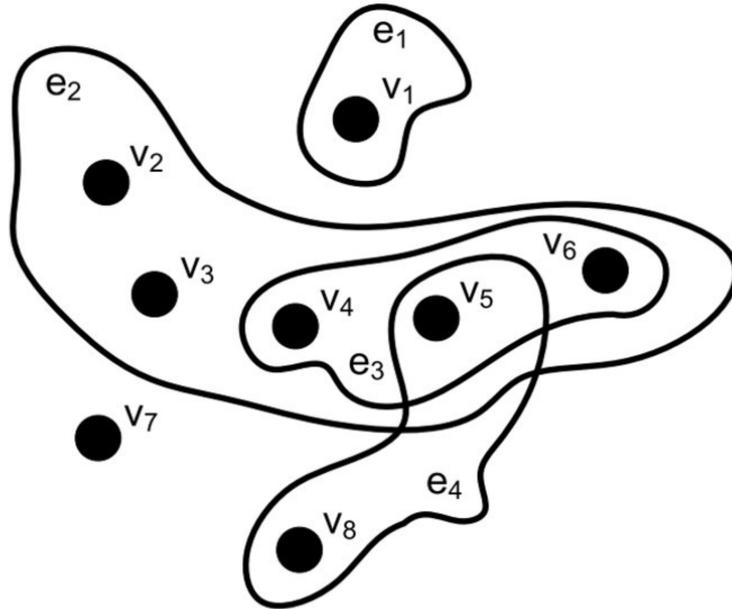


Figure 1: Visualisation of a flexible process graph based on [6]

The visualisation of a flexible process graph can be seen above on Figure 1, where $\{e_1, e_2, e_3, e_4\} \in E$ are the edges and $\{v_1, v_2, v_3, v_4, v_5, v_6, v_7, v_8\} \in V$ are the activity nodes. The following visualisation is an end result of the applied flexible process graph modeling on a Data science MOOC syllabus. Figure 2 shows how easily can this solution represent different learning paths.

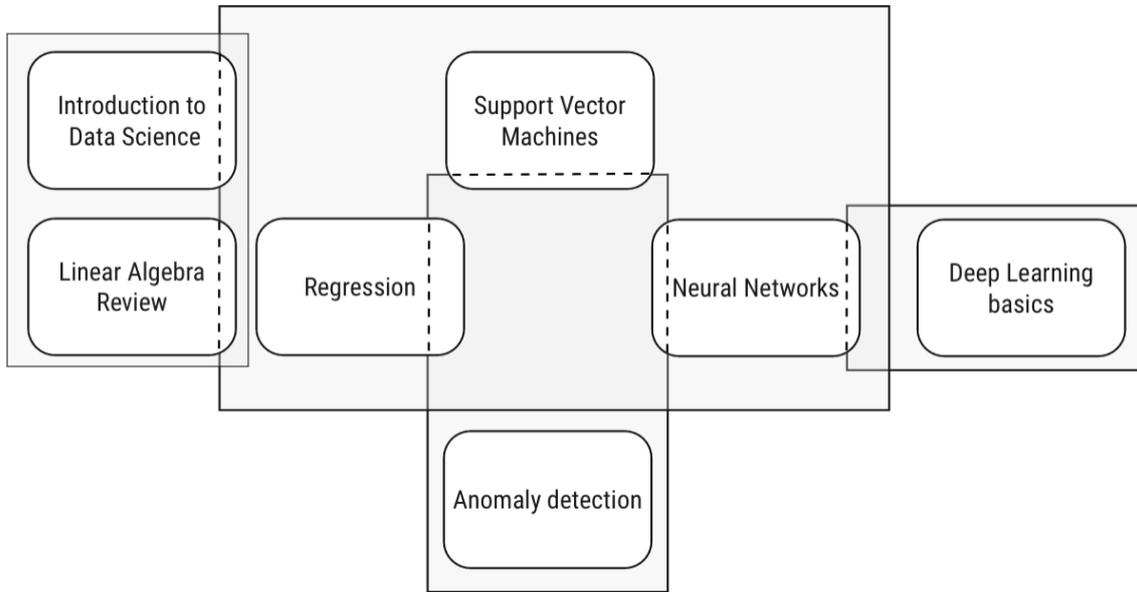


Figure 2: Flexible process graph of "Machine learning basics" MOOC syllabus

3 Conclusion and Future work

Based on the conducted research the most fitting solution for modelling micro-credentials' learning process, which can be an ad-hoc considering the order of activities that the learners can take, is the Flexible Process Graph [6] or one of its variants. The conventional techniques lead to a more difficult to interpret model on which the graph algorithms struggle in an ad-hoc setting.

The flexible process graphs don't have the unified framework for graph algorithms as it is the case in the directed graphs. This leads to our future research, where the transformation of graph algorithms to flexible process graph algorithms will be developed and evaluated.

Next to the graph algorithms, this research raised an other question, which is interesting to research following these results. It is aiming to push the transparency of the learning process modeling and holding their credentials. With the conformance checking algorithms an automatic micro-credential or diploma issuing solution can be built. After building this solution a research will be conducted on how to create an interface with the current blockchain based micro-credential storing architectures like [4, 7] and the algorithm that issues them.

Acknowledgment The project was supported by the European Union, co-financed by the European Social Fund (EFOP-3.6.3-VEKOP-16-2017-00002) and the project was partially supported by „Application Domain Specific Highly Reliable IT Solutions” project that has been implemented with the support provided from the National Research, Development and Innovation Fund of Hungary, financed under the Thematic Excellence Programme TKP2020-NKA-06 (National Challenges Subprogramme) funding scheme.

References

- [1] Tine Andersen, Futures Hanne Shapiro, and Larsen Kristine Nedergaard. A european approach to micro-credentials. URL: <https://op.europa.eu/en/publication-detail/-/publication/7a939850-6c18-11eb-aeb5-01aa75ed71a1>.
- [2] Alfredo Bolt, Massimiliano de Leoni, Wil Aalst, and Pierre Gorissen. Business process reporting using process mining, analytic workflows and process cubes: A case study in education. In *Lecture Notes in Business Information Processing*, volume 244, pages 28–53, 01 2017. doi:10.1007/978-3-319-53435-0_2.
- [3] George Giaglis. A taxonomy of business process modeling and information systems modeling techniques. *International Journal of Flexible Manufacturing Systems*, 13:209–228, 04 2001. doi:10.1023/A:1011139719773.
- [4] Merija Jirgensons and Janis Kapenieks. Blockchain and the future of digital learning credential assessment and management. *Journal of teacher education for sustainability*, 20(1):145–156, 2018.
- [5] Open Group. Togaf: The open group architecture framework, version 9. electronic, 2010. URL: <http://www.opengroup.org/togaf/>, arXiv:<http://www.opengroup.org/togaf/>.
- [6] Artem Polyvyanyy and Mathias Weske. Hypergraph-based modeling of ad-hoc business processes. In Danilo Ardagna, Massimo Mecella, and Jian Yang, editors, *Business Process Management Workshops*, pages 278–289, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg.
- [7] Unique Poudel and Sai Gajjela. *A BLOCKCHAIN BASED MICRO-CREDENTIALING SYSTEM*. PhD thesis, Deakin University, 05 2019. doi:10.13140/RG.2.2.18403.20008.
- [8] John A. Zachman. A framework for information systems architecture. *IBM systems journal*, 26(3):276–292, 1987.

An Operationalized Transformation for Activity Diagram into YAWL

Meriem Kherbouche ¹  0000-0002-4592-1765 Ahmad Mukashaty

Bálint Molnár ²  0000-0001-5015-8883

^{1,2} Information Systems Department, Eötvös Loránd University of Budapest, ELTE IK,
Pázmány Péter sétány 1/C, 1117 Budapest, Hungary

{meriemkherbouche, molnarba}@inf.elte.hu, ahmedmukashaty@gmail.com

Abstract Nowadays, the Unified Modeling Language (UML), standardized by the Object Management Group (OMG) has been widely accepted by the industry and has established itself as the common language for analysis and design in software engineering.

The most known weakness of UML is that it can't be verified directly, a transformation to another more formal language is needed to facilitate the verification process. Model to Program transformation (M2P) is an important step to process and validate UML diagrams that are designed in software modelling environments, and it bridges between languages on a different level of abstraction and formality.

The research outlines an approach of transformation from UML diagrams to a formal workflow language YAWL (Yet Another Workflow Language). This transformation simplifies the semantics of UML diagrams via a mapping to YAWL, by defining a set of transformation rules, which in turn makes the verification and analysis of UML models easier and provides a chance to operationalize the model.

1 Introduction

Model transformation is a mechanism for deriving from one model (source model) to another (destination model) while maintaining some kind of equivalence between them by defining and executing a set of rules known as model transformation language. There are many, various types of model transformations and applications, each with its own set of inputs and outputs, as well as the manner they are stated.

The Unified Modeling Language (UML) is one of the most transformed languages which is defined as a meta-model with several packages. Each package introduces concepts expressed through graphical notation and diagrams. In this paper, we will present a novel transformation from UML activity diagrams models, that can fairly capture protocol designs, to YAWL-Net in both behavioral and functional notions by constructing a set of transformation algorithms and mapping rules. the weakness of UML is that it cannot be verified directly, a transformation to another more formal language is needed to facilitate the verification.

YAWL is semantically based on the Petri net. YAWL have a mature verification tools that find structural errors (WofYAWL [1], Woflan [2], ProM [3]). First, we transform certain components of the UML-AD model respecting the element-to-element rules. Second, UML-AD control flows are converted to yawl split or join gates. Third, get rid of any redundant tasks that aren't adding any value to the process.

In section 2 we present some literature review about process modeling, workflow language, and unified modelling language. YAWL and AD-UML meta-modeling, in section 3 we provide foundations and preliminaries about model transformation, methods and materials used in transformation. Section 4 illustrates the transformation context, the transformation of basic structures, connectors, model reduction, and branch conditions and the algorithm of transformation that could be modeled semantically without the need for programming. After that, a conclusion and future works.

2 Literature Review

2.1 Business Process Modelling

In a variety of fields, business process modeling is used to characterize and state customized process and information system development. Many factors and information are extracted from business process like cost and time, in addition to execution improvement and fault management.

2.1.1 Business Model Life Cycle

There are several stages in the life cycle of a business process. In the Analysis phase, elicitation meetings with stockholders are held to develop the AS-IS process model, which includes assessments of the changes and their costs, as well as improvement deficiencies. Design phase is the second phase where the findings of the analysis phase is considered, and TO-BE model is prepared. This model is in charge of the improvement plan as well as new features and modifications in processes such as inputs, outputs, rules, and actions in order to accomplish desired outcomes (efficiency and effectiveness). Validation (simulation) of business processes can be done after the Design phase but before the Implementation phase. During the implementation phase, various organizational and technical details of the enterprise are realized, such as deployment, organizational structure, and resource allocation, and the requirements are mapped into IT services. In the execution phase, business operations are executed to meet client needs, and the execution data is saved in the form of log files or tables using an information system. The result of the business process execution is used to assess the business process and its components. Evaluation phase is the last phase where different quantitative measures are taken, and actual values are compared to target values in order to assess the performance of business objects and costs, and these data are utilized for qualitative indicators such as customer satisfaction and overall quality.

2.2 Formal Workflow Language

It's a workflow language based on workflow patterns, with a software system that includes an execution engine, a graphical editor, and a worklist handler.

2.2.1 YAWL

This language was built on Petri nets, a well-known theory of concurrent processes with a graphical representation, on the one hand, and Workflow Patterns, on the other. The existence of Workflow Patterns are a widely accepted criterion for a process definition

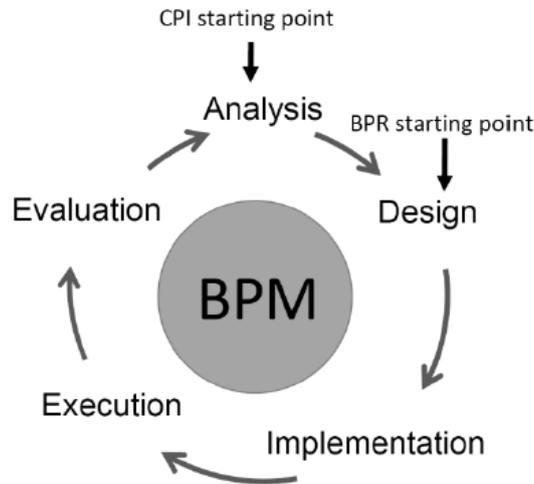


Figure 1: Business model life cycle diagram

language’s applicability. Petri nets can capture many of the described control-flow patterns, but not the many instance patterns, cancellation patterns, or generalized OR-join. YAWL therefore extends Petri nets with dedicated constructs to deal with these patterns.

2.2.2 YAWL Main Features

YAWL offers the following distinctive features:

- For control-flow patterns, YAWL provides extensive support. It is the most powerful language for encapsulating control-flow dependencies in a process design.
- YAWL captures the data perspective through the use of XML Schema, XPath, and XQuery.
- YAWL provides complete resource pattern support. It is the most powerful language for encapsulating resourcing requirements in a process definition.
- YAWL is built on a solid formal foundation. As a result, its specifications are clear and automated verification is possible (YAWL offers two distinct approaches to verification, one based on Reset nets, the other based on transition invariants through the WofYAWL editor plug-in).
- YAWL’s Worklets methodology provides unique support for dynamic workflow. As a result, workflows can change over time to meet new and changing needs.
- YAWL is designed to be simple to set up. It comes with a variety of automatic installers as well as a user-friendly graphical design environment.

2.3 Unified Modelling Language

The primary goal of UML is to establish a standard for visualizing the design of a system. UML diagrams are used to depict a system’s behavior and structure. UML is a model-

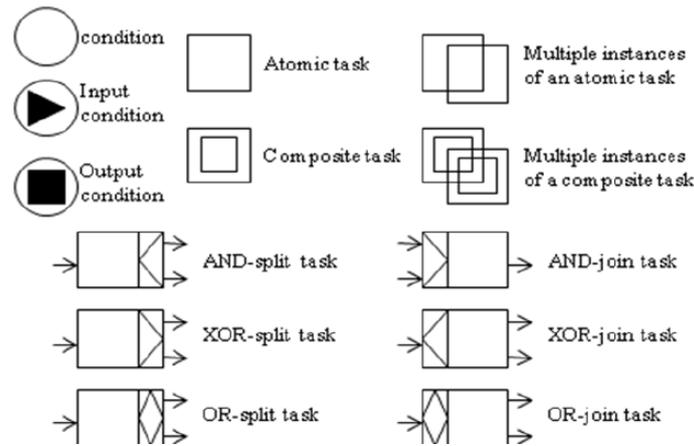


Figure 2: YAWL language main notations

ing, design, and analysis tool that helps software engineers, businesspeople, and system architects.

2.3.1 What is Activity Diagram

The inputs, outputs, sequences, and circumstances for coordinating other behaviors are all highlighted in an activity diagram. Semantics are included in activity diagrams to precisely specify system behavior in terms of control flow, inputs, and outputs. A controlled series of actions that changes inputs into outputs is represented by an activity diagram.

2.3.2 Activity Diagram Meta Modeling Overview

An Activity is a graph with three kinds of ActivityNodes: ObjectNode, ControlNode and ExecutableNode. An ObjectNode represents the data in a process, a ControlNode coordinates the execution flow and an ExecutableNode represents a node that can be executed, i.e. process action. There are two kinds of ActivityEdge to link the nodes: ObjectFlow and ControlFlow. ObjectFlow edges connect ObjectNodes and can have data passing along it. ControlFlow edges constrains the desired order of execution of the ActivityNodes. ControlNode can be used for parallel routing (ForkJoin), conditional routing (DecisionNode), synchronization (JoinNode) and merging multiple alternate flows (MergeNode). InitialNode and AcitivityFinalNode represent respectively the beginning and the end of an Activity while FlowFinalNode terminates flow. InputPin and OutputPin are anchored to Actions to represent the required input data and the output data produced by the action. Similarly, an Activity can have multiple ActivityParameterNode to represent its data input and output. Thus, an Activity can represent a process by defining a coordinated sequencing set of actions using both control-flow and data-flow.

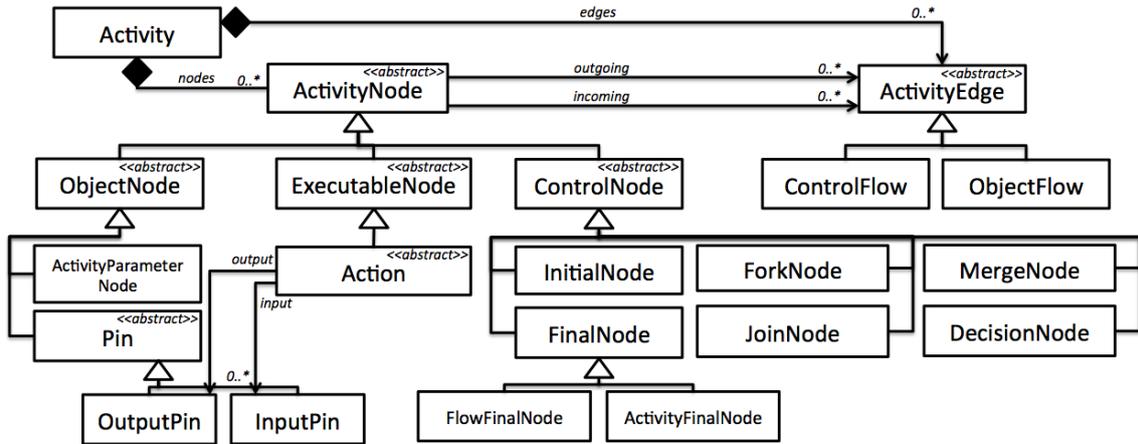


Figure 3: YAWL language main symbols

3 Methods and Materials

3.1 Model Transformation

It is a mechanism for deriving from one model (source model) to another (destination model) while maintaining some sort of equivalence relationship between them by creating and executing a set of rules known as model transformation language.

3.1.1 Types of model transformation

- Model To Text Transformation (M2T): is a type of model transformation in which the result of model transformation is source code or configuration text.
- Text To Model Transformation (T2M): it is a reverse engineering process where the text is transformed into information defines behavioral concepts.
- Text To Text Transformation (T2T): it is an approach used for language processing in order to encode a text and transform it into another different text.
- Model To Model Transformation (M2M): is a model-driven process which enables to specify the source and destination models and set of mapping declarations that define the relationships between the elements in the model.

3.2 Methods & Materials

3.2.1 Papyrus

It is UML graphical editing tool which can be used as an Eclipse plugin which provides support for domain specific languages and SysML.

3.2.2 Acceleo

It is a model to text (M2T) transformation tool which can be used to express transformations and generate a corresponding generated code.

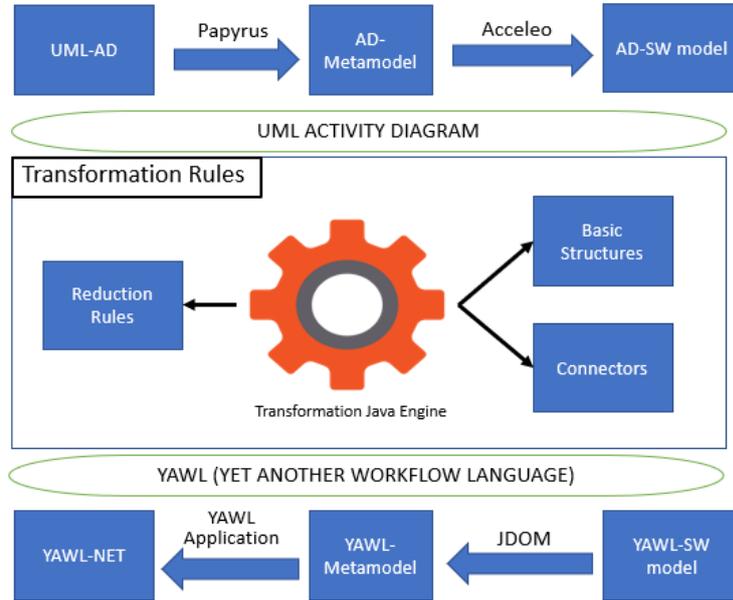


Figure 4: UML-AD to YAWL-NET Transformation Context

3.2.3 JDOM

It is an open-source Java-based document object model for XML that was designed specifically for the Java platform in that it can take advantage of its language features. It uses external parsers to build documents.

4 Research Scope

4.1 Transformation Context

In order to achieve model-to-model transformation, which is the transformation from UML-AD to YAWL-NET, the transformation process should pass through different stages of transformation and each stage has a set of interactions to implement. The source is an activity diagram built on Papyrus UML, which provides services to export the abstract syntax of an activity diagram and its corresponding XML format. The output is an XML (.yawl) file that can be parsed and run and interpret by YAWL Application.

4.2 UML-AD Metamodel to SW-Model transformation

Metamodel to software model transformation is a refactoring method which generates integrated classes in order to communicate with the transformation engine module. Model to Text languages is performed using Acceleo Tool, it mainly reads UML-AD XML file, generated by Papyrus, and auto generate one Enum (Node Type), three classes (Edge, Node, Activity), and two functions (Node Constructor & Activity Constructor).

4.3 Transformation Rules from UML-AD to YAWL-NET

Transformation engine is the heart of the transformation chain, in this stage, transformation behavioral and rules are defined, and suitable transformation algorithm is performed.

4.3.1 Transformation of basic structure

An element-to-element mapping rules are implemented on some of UML-AD elements which has a corresponding element in YAWL-NET, and both of them have the same behavior and functionality on the net.

- Initial Node in UML-AD matches Input Condition in YAWL-NET.
- Activity Final Node in UML-AD matches Output Condition in YAWL-NET.
- Opaque Action in UML-AD matches Task in YAWL-NET.

4.3.2 Transformation of connectors

Defining mapping rules from control flows in UML-AD (fork, decision, merge, and join) to YAWL NET gates (XOR, AND) which is assigned to YAWL atomic task. In YAWL connectors are part of tasks, in order to map a connector, an empty task (virtual task) is defined to represent a single connector.

- Decision node is transformed to XOR-Split task.
- Merge node is transformed to XOR-Join task.
- Fork node is transformed to AND-Split task.
- Join node is transformed to AND-Join task.

4.3.3 Model Reduction

In order to avoid any redundant empty tasks in the output YAWL nets after transforming the activity control flows, reduction methods are implemented on some empty tasks which is considered as duplicated ones.

4.3.4 Define Branch Conditions

When it comes to decision symbol transformation, the main concept that the decision depends on dimensions in order to choose one of multi branches. In order to preserve that concept, we defined a condition as an input of each branch, which is expected to be a Boolean function that allow the token to pass in case of true.

5 Conclusion

In this research, we presented a novel transformation from UML-AD to YAWL-Net by implementing a set of transformation algorithms. First, it applies an element to element mapping rules for the basic structure. Second, transforming UML-AD control flows to yawl

split or join gates. Third, remove any redundant task that doesn't have any additional functionality to the process.

In future research, UML-AD model must be analyzed with YAWL verification tools such as (Woflan, WofYawl) to check whether it contains structural errors like deadlock or lack of synchronization.

Acknowledgment The project was supported by the European Union, co-financed by the European Social Fund (EFOP-3.6.3-VEKOP-16-2017-00002) and the project was partially supported by „Application Domain Specific Highly Reliable IT Solutions” project that has been implemented with the support provided from the National Research, Development and Innovation Fund of Hungary, financed under the Thematic Excellence Programme TKP2020-NKA-06 (National Challenges Subprogramme) funding scheme.

References

- [1] Messaoud Abbas, Choukri-Bey Ben-Yelles, and Renaud Rioboo. Modelling uml state machines with focalize. *International Journal of Information and Communication Technology*, 13(1):34–54, 2018. doi:10.1504/ijict.2018.10010449.
- [2] Kyriakos Anastasakis, Behzad Bordbar, Geri Georg, and Indrakshi Ray. Uml2alloy: A challenging model transformation. In *International Conference on Model Driven Engineering Languages and Systems*, pages 436–450. Springer, 2007.
- [3] Kyriakos Anastasakis, Behzad Bordbar, Geri Georg, and Indrakshi Ray. On challenges of model transformation from uml to alloy. *Software & Systems Modeling*, 9(1):69, 2010.
- [4] Mira Balaban, Phillipa Bennett, Khanh-Hoang Doan, Geri Georg, Martin Gogolla, Igal Khitron, and Michael Kifer. A comparison of textual modeling languages: Ocl, alloy, foml. In *OCL@ MoDELS*, pages 57–72, 2016.
- [5] Javier Cámara, Carlos Canal, Javier Cubo, and Antonio Vallecillo. Formalizing wsbpel business processes using process algebra. *Electronic Notes in Theoretical Computer Science*, 154(1):159–173, 2006. doi:10.1016/j.entcs.2005.12.038.
- [6] Alcino Cunha, Ana Garis, and Daniel Riesco. Translating between alloy specifications and uml class diagrams annotated with ocl. *Software & Systems Modeling*, 14(1):5–25, 2015.
- [7] Alberto Rodrigues Da Silva. Model-driven engineering: A survey supported by the unified conceptual model. *Computer Languages, Systems & Structures*, 43:139–155, 2015. doi:10.1016/j.cl.2015.06.001.
- [8] Loïc Gammaitoni. *On the Use of Alloy in Engineering Domain Specific Modeling Languages*. PhD thesis, University of Luxembourg, Luxembourg, 2017.
- [9] Loïc Gammaitoni and Pierre Kelsen. F-alloy: An alloy based model transformation language. In *International Conference on Theory and Practice of Model Transformations*, pages 166–180. Springer, 2015. doi:10.1007/978-3-319-21155-8_13.

- [10] Loïc Gammaitoni and Pierre Kelsen. F-alloy: a relational model transformation language based on alloy. *Software & Systems Modeling*, 18(1):213–247, 2019.
- [11] Loïc Gammaitoni, Pierre Kelsen, and Qin Ma. Agile validation of model transformations using compound f-alloy specifications. *Science of Computer Programming*, 162:55–75, 2018. doi:10.1016/j.scico.2017.07.001.
- [12] Ana Garis, Ana CR Paiva, Alcino Cunha, and Daniel Riesco. Specifying uml protocol state machines in alloy. In *International Conference on Integrated Formal Methods*, pages 312–326. Springer, 2012. doi:10.1007/978-3-642-30729-4_22.
- [13] Zhaogang Han, Li Zhang, Jiming Ling, and Shihong Huang. Control-flow pattern based transformation from uml activity diagram to yawl. In *International IFIP Working Conference on Enterprise Interoperability*, pages 129–145. Springer, 2012. doi:10.1007/978-3-642-33068-1_13.
- [14] Edward Huang, Leon F McGinnis, and Steven W Mitchell. Verifying sysml activity diagrams using formal transformation to petri nets. *Systems Engineering*, 2019. doi:10.1002/sys.21524.
- [15] Daniel Jackson. Alloy: a language and tool for exploring software designs. *Communications of the ACM*, 62(9):66–76, 2019. doi:10.1145/3338843.
- [16] Mohammed Hamouda Karboos. Integrating business process concepts into uml activity model. *Journal of Engineering and Computer Science (JECS)*, 19(1):57–68, 2019.
- [17] Oliver Kautz, Shahar Maoz, Jan Oliver Ringert, and Bernhard Rumpe. Cd2alloy: a translation of class diagrams to alloy. *Techn. Rep. AIB-2017-06, RWTH Aachen University (July 2017)*, 2017.
- [18] Meriem Kherbouche, Khawla Bouafia, and Balint Molnar. Transformation of uml state machine to yawl. In *Ninth IEEE International Conference on Intelligent Computing and Information Systems*, 2019. doi:10.1109/icicis46948.2019.9014793.
- [19] Ahsanun Naseh Khudori and Tri Astoto Kurniawan. Business process model transformation techniques: A comprehensive survey. *Advanced Science Letters*, 24(11):8606–8612, 2018. doi:10.1166/asl.2018.12311.
- [20] Kevin Lano and Shekoufeh Kolahdouz-Rahimi. Model transformation specification and design. In *Advances in Computers*, volume 85, pages 123–163. Elsevier, 2012. doi:10.1016/b978-0-12-396526-4.00003-5.
- [21] Mónica A López-Campos, Adolfo Crespo Márquez, and Juan F Gómez Fernández. Modelling using uml and bpmn the integration of open reliability, maintenance and condition monitoring management systems: an application in an electric transformer system. *Computers in industry*, 64(5):524–542, 2013. doi:10.1016/j.compind.2013.02.010.

- [22] Shahar Maoz, Jan Oliver Ringert, and Bernhard Rumpe. Cd2alloy: Class diagrams analysis using alloy revisited. In *International Conference on Model Driven Engineering Languages and Systems*, pages 592–607. Springer, 2011. doi:10.1007/978-3-642-24485-8_44.
- [23] Steffen Mazanek and Mark Minas. Transforming bpmn to bpel using parsing and attribute evaluation with respect to a hypergraph grammar, 2009. URL: http://is.tm.tue.nl/staff/pvgorp/events/grabats2009/submissions/grabats2009/_submission/_8.pdf. (Last accessed on October 14, 2009).
- [24] Bálint Molnár. Applications of hypergraphs in informatics: A survey and opportunities for research. *Annales Universitatis Scientiarum Budapestinensis de Rolando Eotvos Nominatae Sectio Computatorica*, 42:261–282, 2014.
- [25] Anantha Narayanan and Gabor Karsai. Verifying model transformations by structural correspondence. *Electronic Communications of the EASST*, 10, 2008. doi:<http://dx.doi.org/10.14279/tuj.eceasst.10.157>.
- [26] Jan Recker and Marcello La Rosa. Understanding user differences in open-source workflow management system usage intentions. *Information Systems*, 37(3):200–212, 2012. doi:10.1016/j.is.2011.10.002.
- [27] Monika Singh, AK Sharma, and Ruhi Saxena. Formal transformation of uml diagram: Use case, class, sequence diagram with z notation for representing the static and dynamic perspectives of system. In *Proceedings of International Conference on ICT for Sustainable Development*, pages 25–38. Springer, 2016. doi:10.1007/978-981-10-0135-2_3.
- [28] Yentl Van Tendeloo and Hans Vangheluwe. Discrete event system specification modeling and simulation. In *Proceedings of the 2018 Winter Simulation Conference*, pages 162–176. IEEE Press, 2018. doi:10.1109/wsc.2018.8632372.
- [29] Matthias Weidlich, Gero Decker, Alexander Großkopf, and Mathias Weske. Bpel to bpmn: The myth of a straight-forward mapping. In *OTM Confederated International Conferences "On the Move to Meaningful Internet Systems"*, pages 265–282. Springer, 2008. doi:10.1007/978-3-540-88871-0_19.
- [30] Karim Zarour, Djamel Benmerzoug, Nawal Guermouche, and Khalil Drira. A systematic literature review on bpmn extensions. *Business Process Management Journal*, 2019. doi:10.1108/bpmj-01-2019-0040.

An overview of reinforcement learning applications in the control system of the intelligent transportation system

Zhang Yinghong¹ Bálint Molnár²  0000-0001-5015-8883

¹ Data Science and Engineering Department, Eötvös Loránd University of Budapest, ELTE IK, Pázmány Péter sétány 1/C, 1117 Budapest, Hungary
em7qol@inf.elte.hu

² Information Systems Department, Eötvös Loránd University of Budapest, ELTE IK, Pázmány Péter sétány 1/C, 1117 Budapest, Hungary
molnarba@inf.elte.hu

Abstract. With the continuous improvement of global economic integration, the pace of global urbanization and modernization has been accelerated. And with the agricultural and industrial revolutions, the urban population began to expand at an unprecedented rate. Today, 55% of the world's population lives in urban areas, a proportion that is expected to increase to 68% by 2050, additional 2.5 billion city dwellers worldwide. Along with the physical expansion of cities, there are many problems, such as water management problems, social order problems, traffic problems. Among them like the blood of the city traffic, more and more become an obstacle to urban development. Solving the problem of urban traffic congestion and optimize the efficiency of urban traffic is an essential link in building a smart city. Intelligent transportation is an important part of a smart city. How to scientifically design and build intelligent transportation is a very valuable problem. Reinforcement learning, as an important part of the Computer Science, has been widely used in various social fields such as Health, transportation, education, finance, and so on. This paper will discuss the application of intelligent transportation control systems through the investigation and research of intelligent transportation combined with reinforcement learning.

Keywords. Intelligent transportation system (ITS), smart city, reinforcement learning, deep learning, deep reinforcement learning, internet of things, IoT smart service, 5G communications, control systems.

1 Introduction to ITS

Along with the physical expansion of cities, there are many problems, such as water management problems, social order problems, traffic problems. Among them, the blood of the city traffic becomes more and more an obstacle to urban development. The emergence of intelligent transportation can optimize traffic to some extent and improve urban traffic problems.

Intelligent Transportation Systems (ITS) apply various technologies to monitor, evaluate, and manage transportation systems to enhance efficiency and safety [1]. Science fiction transportation aside for the moment, this definition can be reduced to the following concepts that make up intelligent transportation: management, efficiency, and safety. In other words, it uses emerging technologies to make urban mobility more convenient, cheaper, and safer for governments and individuals.

Emerging technologies are taking these ideas from possible to common. Mainly due to the proliferation of IoT devices and 5G communication technology. The former provides inexpensive sensors and controllers that can be embedded into virtually any physical machine for remote control and management. The latter gives high-speed communications needed for real-time management and control of transportation systems with minimal latency [2].

2 ITS control systems with reinforcement learning

In urban intelligent transportation, pedestrians and vehicles (especially cars) are the main moving objects. However, it is often cars rather than people that move long distances in cities, so here we take cars as the main research object in urban intelligent transportation or study the ITS control system with cars as the boundary.

On this basis, we can consider the practical application of reinforcement learning from the outside and inside of the car, that is, reinforcement learning can be applied to ITS control system from the outside and inside of the car.

In this way, we can divide the ITS control system into external and internal. The external mainly refers to the traffic light control system that affects the driving of the car, and the internal mainly refers to the braking system that affects the driving of the car. Here we mainly discuss the application of reinforcement learning in ITS control system from these two major perspectives.

2.1 Internal part of ITS control system

The braking system is the core of the vehicle driving control system, which determines the speed of the vehicle and the safety of the passengers. When we take the vehicle as the research object of the ITS control system, to a certain extent the main control system of the vehicle itself can be used as the internal part of the ITS control system.

Most traditional automatic braking systems are based on rules, which specify specific braking control protocols for different situations. But on real roads, various situations may occur. In other words, the braking rules formulated in advance are likely to fail. This shows that the rule-based braking method has limitations.

In ITS, autonomous driving is an important part and is the technological development trend of ITS in the future. When we consider the internal part of the ITS control system, we should proceed from the perspective of autonomous driving. Different from the human-controlled braking system, the automatic braking system of autonomous driving is more autonomous and can be discussed as an internal part of the ITS control system. The automatic braking system is the soul of achieving safe and autonomous driving. When a threatening obstacle is detected, the system can automatically reduce the speed of the vehicle. Automatic braking should provide safe and comfortable braking control, without premature or late braking performance.

In recent years, people's interest in machine learning and reinforcement learning has exploded. Especially deep neural network (DNN) technology has been widely used in autonomous driving technology. Reinforcement learning (RL) technology has also been significantly improved with the adoption of deep neural network (DNN) technology. This technique, called Deep Reinforcement Learning (DRL), performs quite well on various challenging robotics and control problems. In [3], Deep Q-network (DQN) DRL technology

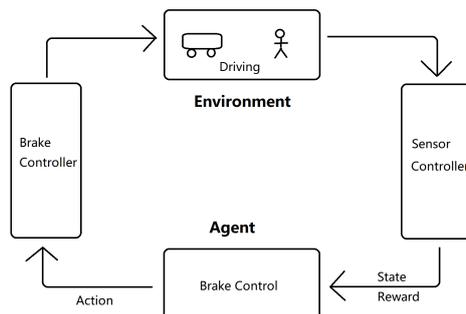


Figure 1: A simple example of autonomous braking systems based on DRL.

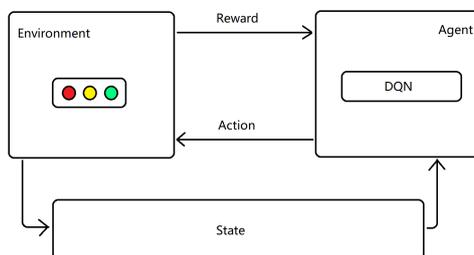


Figure 2: A simple framework for traffic light control based on DRL.

was proposed, which uses DNN to approximate q-valued functions. The results show that DQN is superior to human experts in various Atari video games. In recent years, DRL has been applied to the control systems of autonomous vehicles, such as in Reference [4, 5]. A simple example of autonomous braking systems based on DRL is shown in Figure 1.

2.2 External part of ITS control system

Although autonomous driving is the future development trend of ITS, it will take a long time to realize Level 5 autonomous driving (fully autonomous driving). Compared with autonomous driving, optimizing traffic lights in the traditional sense is more cost-effective and easier to implement. Based on the above, we also take the vehicle (car) as the research object. Here, we will discuss the representative application example of reinforcement learning, intelligent traffic light control, from the outside of the ITS control system.

At present, most traffic lights are still controlled by a predetermined fixed time plan [6], not designed by observing actual traffic. Recent studies have proposed manual rules based on real traffic data [7]. However, these rules are still predefined, and real-time traffic cannot be dynamically adjusted. Although most of the existing traffic lights are operated by manual rules, the intelligent traffic light control system should be dynamically adjusted to adapt to real-time traffic. Thanks to the rise of machine learning, the use of deep reinforcement learning technology for traffic light control is an emerging trend. A simple framework for traffic light control based on DRL is shown in the Figure 2.

3 Conclusion

With the acceleration of the urbanization process, more and more people move to the city, and with the expansion of the city, there are also various problems affecting human production and life, especially the core traffic problem of the city. The development of science and technology makes it possible for an intelligent transportation system to solve traffic problems. However, as an important part of intelligent transportation, the control mode should be paid more attention to and optimized.

In this paper, two kinds of internal and external control modes, which influence the intelligent transportation system, are discussed in the field of reinforcement learning, considering the vehicle as to the main analysis object.

Acknowledgment The project was supported by the European Union, co-financed by the European Social Fund (EFOP-3.6.3-VEKOP-16-2017-00002) and the project was partially supported by „Application Domain Specific Highly Reliable IT Solutions” project that has been implemented with the support provided from the National Research, Development and Innovation Fund of Hungary, financed under the Thematic Excellence Programme TKP2020-NKA-06 (National Challenges Subprogramme) funding scheme.

References

- [1] U.S. Department of Transportation (US DOT) 1200 New Jersey Avenue, SE Washington, DC 20590 800.853.1351. *ITS Professional Capacity Building Program*. <https://www.pcb.its.dot.gov/eprimer/default.aspx>
- [2] Steve Mazur, Business Development Director, Government. (December 09, 2020) *An Introduction to Smart Transportation: Benefits and Examples*. <https://www.digi.com/blog/post/introduction-to-smart-transportation-benefits>
- [3] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg and D. Hassabis, "Human-level control through deep reinforcement learning," *Nature*, vol. 518(7540), pp. 529-533, 2015.
- [4] J. Koutnik, J. Schmidhuber and F. Gomez, "Evolving deep unsupervised convolutional networks for vision-based reinforcement learning," *Proceedings of the 2014 Annual Conference on Genetic and Evolutionary Computation*, ACM, pp. 541-548, 2014.
- [5] C. Desjardins and B. Chaib-draa, "Cooperative adaptive cruise control: a reinforcement learning approach," *IEEE Trans. Intelligent Transportation Systems*, vol. 12, no. 4, pp. 1248-1260 Dec. 2011.
- [6] Alan J Miller. 1963. Settings for fixed-cycle traffic signals. *Journal of the Operational Research Society* 14, 4 (1963), 373-386
- [7] Isaac Porche and Stéphane Lafortune. 1999. Adaptive look-ahead optimization of traffic signals. *Journal of Intelligent Transportation System* 4, 3-4 (1999), 209-254.

Credit risk management in financial services

Ekaterina Zolotareva, Bethelihem Seifu,
Bálint Molnár  0000-0001-5015-8883

Information Systems Department, Eötvös Loránd University of Budapest, Pázmány Péter
sétány 1/C, 1117 Budapest, Hungary
(molnarba,dnbo45,c7nbxt)@inf.elte.hu

Since Risk management is the identification, evaluation, and management of threats to the capital and income of an organization, threats or risks may come from a variety of sources, including financial insecurity, legal liabilities, policy errors, accidents, and disasters. In digitized enterprises, IT security threats and data-related risks, and risk management strategies have been made a top priority. As a result, the risk management plan increasingly includes companies' identification and monitoring processes, including proprietary corporate data, personal information, and intellectual data, for threats to their digital assets, Including private corporate data, customer identity, and intellectual property information. Each company and organization, in the event of unforeseen, harmful events, can cost or cause the company to close permanently. Risk management is still a problem for most companies nowadays due to different reasons such as failure to use appropriate risk metrics, the mismeasurement of known risks, failure to take known risks into account, and failure in monitoring and managing risks. During our research, we have used a Hungarian well-known bank data-set which is OTP bank to analyze the customer specifically companies' risks given different features of them. We have used different state-of-art or cutting edge models such as xgboost, catboost and Knn algorithms which are useful for future prediction of risk level associated with the companies. After cleaning the data-set and using the different classification algorithms, we came up with comparably good results which we measured with accuracy evaluation metrics.

1 Introduction

Banks, like other companies, are looking for ways to manage their risks while simultaneously seeking to increase productivity and efficiency to create value. This performance is only achieved when banks issue loans to customers from money deposited by shareholders or customer savings, thereby exposing them to risk in the event of default. Despite this risk, banks cannot stop providing loans, as this is the main source of their profitability. Thus, they find themselves in a situation to achieve profitability, on the one hand, and the risk of default, on the other hand. To achieve success, the only option is good credit risk management practices, since, in this process, returns are correlated with risk. Because of the importance and relevance of this area, there is still room for improvement and testing of SOTA approaches. The greatest innovations revolution in the world was ever created by AI and specifically machine learning. It offers great opportunities in the financial sector to increase customer experience, guarantee consumer protection and improve customer risk management significantly.

Although the implementation of state-of-the-art machine learning models is easier than ever, it is challenging to design and implement systems that support real-world financial applications. Since the global financial crisis, risk management in banks has become more prominent. Thereby, the detection, measurement, reporting, and management of risks have been a continuous focus. This paper aims to analyze and evaluate those machine-learning techniques that are researched within the context of banking risk management and make a predictive analysis of customers. Companies being partners of a Hungarian bank are categorized at a minimum, average, and high risk based on their historical financial status. The categorization is based on the capital, the taxes paid, the net sales revenue, the current state, and other criteria. We have used different supervised learning models to classify the companies' risk levels so that we have chosen those algorithms that have the highest accuracy and performance. We discuss the results in detail and present multiple comparisons using a table that includes all machine learning models that were used and their accuracy in different versions respectively

2 Related Works

Credit risk is considered as the chance of loss that will occur when the loan or any other line of credit by a particular debtor is not repaid (Campbell, 2007) [1]. Since 2008, financial experts around the world have researched and analyzed the primary factors underpinning the credit crisis to identify problematic behavior and effective solutions that can help financial institutions avoid catastrophe in the future. Long ago, the Basel Committee on Banking Supervision (1999) has also identified credit risk as a potential threat to the banking sector and developed certain banking regulations that must be maintained by the banks around the world. Owojori, Akintoye, and Adidu (2011)[2] stated that there are legislative inadequacies in the financial system especially in the banking system that are in effect as well as lack of uniform credit information sharing amongst banks cause problems. Thus, the authors urge the fact that banks need to emphasize better risk management strategies that may protect them in the long run.

Kou, Chao, Peng, Alsaadi and Herrera-Viedma, (2019) [3] identified that financial systemic risk is a major issue in financial systems and economics. Machine learning methods are employed by researchers that are trying to respond to systemic risks with the help of financial market data. Machine learning methods are used for understanding the outbreak and contagion of the systemic risk for improving the current regulations of the financial market and industry. The paper studies the research and methodologies on measurement of financial systemic risk with the help of big data analysis, sentiment analysis, and network analysis. Machine learning methods are used along with systemic financial risk management for controlling the overall risks faced by the banks that are related to hedging of the financial instruments of the bank (Kou, Chao, Peng, Alsaadi, Herrera-Viedma, 2019) [3].

3 Problem Statement

As part of this work, we were given anonymized data, which is described in section 4. The purpose of this work is to perform experiments using various machine learning models to obtain the best accuracy in predicting which group a company belongs to a certain

category. During the course of our work, we encountered the following challenges:

1. Highly imbalanced number of samples within the classes
2. A small number of features environment.
3. Lots of noisy data

4 Dataset

We use the Opten Corp data set for the experimentation, obtained from a Hungarian bank. It contains 39,3737 records with 11 company features. The objective is to build a supervised classifier that has high accuracy in distinguishing single companies between the three classes (i.e. high, medium, or low risk).

The bank dataset consists of 11 features, ranging from the company name to the financial status of the company such as tax paid, capital, and others. Based on the available information, our target was to classify companies based on their current financial status by using the more important features. The dataset had missing values for the features net sales revenue, tax paid, capital, and rating with 1004, 1169, 376, 6890 in number respectively.

5 System

In this work we have implemented the most common and novel models for risk management task, the architecture of each of which will be discussed in details as well as model implementation information. As the additional step we have implemented very general algorithms to compare them with SOTA approaches.

6 Evaluation

For evaluating metrics the accuracy was used. The accuracy of a machine learning classification algorithm is one way to measure how often the algorithm classifies a data point correctly [4]. Accuracy is the number of correctly predicted data points out of all the data points. More formally, it is defined as the number of true positives and true negatives divided by the number of true positives, true negatives, false positives, and false negatives. A true positive or true negative is a data point that the algorithm correctly classified as true or false, respectively. A false positive or false negative, on the other hand, is a data point that the algorithm incorrectly classified. For example, if the algorithm classified a false data point as true, it would be a false positive. Often, accuracy is used along with precision and recall, which are other different metrics that use various ratios of true/false, and positives/negatives results. Together, these metrics provide a detailed look at how the algorithm is classifying data points.

7 Results

The experimental results of the models were compared with each other in the resulting table. According to the experimental results presented, the tuned SOTA model outperformed

the other models in accuracy estimation.

8 Conclusion

In this paper, we have dealt with the risk management task. We used SOTA approaches and implemented very general algorithms with specialization on financial data . Experiments were carried out to verify the effectiveness of the proposed method. Experimental results on the dataset showed that the tuned *catboost* model performed well in the risk management task. The future direction of the research is to fit the model on a new large-scale financial open-source dataset.

9 Acknowledgment

The project was supported by the European Union, co-financed by the European Social Fund (EFOP-3.6.3-VEKOP-16-2017-00002) and the project was partially supported by „Application Domain Specific Highly Reliable IT Solutions” project that has been implemented with the support provided from the National Research, Development and Innovation Fund of Hungary, financed under the Thematic Excellence Programme TKP2020-NKA-06 (National Challenges Subprogramme) funding scheme.

References

- [1] **Campbell A**, "*Bank insolvency and the problem of nonperforming loans.*", Comput. **25-44**, 2007
- [2] **Adekunle A. Owojori, Ishola R. Akintoye and Felix A. Adidu**, "*The challenge of risk management in Nigerian banks in the post consolidation era*", Digital Libraries, 2011
- [3] **Kou, Chao X, Peng Y, Alsaadi FE, Herrera-Viedma E**, "*Machine learning methods for systemic risk analysis in financial sectors.*" Technol Econ Dev Econ, Risk Analysis, 2019
- [4] **BS ISO 5725-1**, "*Accuracy (trueness and precision) of measurement methods and results - Part 1.*" General principles and definitions.", p.1 (1994)

Section:
Neural networks and differential equations

Organizer: Péter L. Simon

Invited talk:

Ferenc Izsák: Adaptive numerical approximation of two-point boundary value problems: a neural network-based approach

Contributions:

- Petra Csomós and Ferenc Izsák: In search of an appropriate loss function for differential equations' initial value problems
- Domonkos Haffner and Ferenc Izsák: Solving the Laplace equation by using neural networks
- Gábor Hidy: Residual neural networks as numerical approximations of differential equations
- András Molnár, Imre Fekete and Péter L. Simon: Learning a function from data by solving a differential equation and tuning its parameters
- Anita Windisch: Saddle-node bifurcation in a 3-dimensional neural network model



Adaptive numerical approximation of two-point boundary value problems: a neural network-based approach

Ferenc Izsák

Department of Applied Analysis and Computational Mathematics & AI ELTE Research Group, Eötvös Loránd University, Pázmány P. stny. 1.C, 1117 Budapest, Hungary

`ferenc.izsak@ttk.elte.hu`

Abstract

An adaptive finite element method is developed here for the numerical solution of one-dimensional boundary value problems. The method is based on a neural network representation of continuous, piecewise linear functions. The proposed optimization procedure is demonstrated in a test problem.

1 Introduction

Neural networks have proven their usefulness in a wide range of scientific computing. For classical problems in the numerical analysis, its application is less usual. In this contribution, we propose a way to contribute to this research direction.

For the formal introduction of neural networks, we refer to [3] and [4]. For our purpose, it is sufficient to use that one can assign to any neural network NN a real vector function $\mathcal{NN} : \mathbb{R}^{d_0} \rightarrow \mathbb{R}^{d_N}$, which maps to the input an output value. The function itself is given in concrete terms using some internal parameters. The main power of this approach lies in the efficient optimization procedure, which drives the optimal choice of these model parameters. For this, we mostly use given input-output pairs and choose parameters, which lead to smallest deviation between computed and known outputs. In the absence of given pairs, we can also try to define a meaningful loss function, which should be minimized directly to get optimal parameters.

The automatic differentiation in the related program packages makes possible to deal also with millions of parameters. At the same time, we should not misuse this capability and keep the number of parameters at a moderate level to avoid overfitting and enhance the computational efficiency.

Accordingly, we use here the idea to solve a problem in numerical analysis by converting it into a multidimensional minimization.

2 Problem statement and methods

We investigate two-point boundary value problems for second-order ordinary differential equations of the following form:

$$\begin{cases} -u''(x) + c(x)u(x) = f(x) & x \in (a, b) \\ u(a) = u(b) = 0, \end{cases} \quad (1)$$

where $c \in L_\infty(\Omega)$ and $f \in L_2(\Omega)$ are given.

We are looking for a numerical approximation $u_h : [a, b] \rightarrow \mathbb{R}$ of u as a piecewise linear, continuous function. In concrete terms, we assume that they are linear on $[t_j, t_{j+1}]$ with the slope s_j , where $t_0 = a, t_{N+1} = b$. Such a function can be characterized with $[t_1, t_2, \dots, t_N]$ and $[s_1, s_2, \dots, s_N]$.

According to [4], such a function x_h can be identified with the neural network

$$x_h(t) := \mathcal{NN}(t) = \mathbf{a}_2 \cdot \text{ReLu}(\mathbf{a}_1 t + \mathbf{b}_1) + b_2, \quad (2)$$

with the input t , and parameters on the first layer

$$\mathbf{a}_1 = (1, 1, \dots, 1) \in \mathbb{R}^{N+1} \quad \text{and} \quad \mathbf{b}_1 = (0, -t_1, \dots, -t_N) \in \mathbb{R}^{N+1}$$

and on the second layer

$$\mathbf{a}_2 = (s_1, s_2 - s_1, \dots, s_N - s_{N-1}, s_{N+1} - s_N) \in \mathbb{R}^{N+1} \quad \text{and} \quad b_2 = x_0 \in \mathbb{R},$$

respectively. Note that here s_{N+1} is a known parameter. For the details, see [4].

To find optimal parameters in the above setting, we cannot use known input-output pairs. Instead, the following statement delivers an appropriate loss function.

Theorem 1 *The function $u \in H_0^1(a, b)$ is the unique solution of (1) if and only if $u \in H_0^1(\Omega)$ is the unique minimum of $J : H_0^1(a, b) \rightarrow \mathbb{R}$:*

$$J(u) = \frac{1}{2} \int_a^b (u')^2 + cu^2 - \int_a^b f \cdot u. \quad (3)$$

In this way, our approach is to find parameters $\mathbf{t} = (t_1, t_2, \dots, t_{N-1}) \in \mathbb{R}^{N-1}$ and $\mathbf{s} = (s_1, s_2, \dots, s_{N-1}) \in \mathbb{R}^{N-1}$ such that $J(u_{\mathbf{s}, \mathbf{t}}) := J_{\mathbf{s}, \mathbf{t}}$ is minimal, where $u_{\mathbf{s}, \mathbf{t}}$ denotes the piecewise linear function described at the beginning of the section.

To optimize the performance of our algorithm, we use the following statement.

Lemma 2 *For any fixed parameter set \mathbf{t} above, the minimum of $J_{\mathbf{s}, \mathbf{t}}$ is attained, if the corresponding function $u_{\mathbf{s}, \mathbf{t}}$ is the finite element solution of (1) using a piecewise first order basis with internal vertices $t_1 \leq t_2 \leq \dots \leq t_N$.*

For the proof of the above two statements, we refer to [1]. Using these results, we can reduce the number of parameters in the minimization problem and consider henceforth the following problem:

Find the parameter \mathbf{t} such that $J_{\mathbf{s}(\mathbf{t}), \mathbf{t}}$ is minimal, where $\mathbf{s}(\mathbf{t})$ corresponds to the finite element solution in Lemma 2.

Observe that this is, indeed, an adaptive finite element algorithm, where the basis points t_1, t_2, \dots, t_N are to find in an optimal way.

It is important to ensure that we have an optimal parameter set also in the discrete case, which is stated in the following:

Lemma 3 *For any fixed N , we have \mathbf{t} and \mathbf{s} above such that $J_{\mathbf{s}, \mathbf{t}}$ is minimal.*

Proof:

According to Lemma 2, it is sufficient to ensure the existence of $\mathbf{t} \in \mathbb{R}^N$, for which $J_{\mathbf{s}(\mathbf{t}),\mathbf{t}}$ is minimal. Since the mappings $\mathbf{t} \rightarrow \mathbf{s}(\mathbf{t})$ and $(\mathbf{t}, \mathbf{s}) \rightarrow J_{\mathbf{t},\mathbf{s}}$ are continuous, the same applies for $\mathbf{t} \rightarrow J_{\mathbf{s}(\mathbf{t}),\mathbf{s}}$. On the other hand, indeed the definition domain of this mapping is

$$\mathcal{T}_N = \{\mathbf{t} = (t_1, t_2, \dots, t_N) : a \leq t_1 \leq t_2 \leq \dots \leq t_N \leq b\} \subset \mathbb{R}^N,$$

which is compact, and therefore, we really have a local minimum at some $\mathbf{t} \in \mathcal{T}_N$. \square

Observe, if the minimum is attained at the boundary of \mathcal{T}_N , then $t_j = t_{j+1}$ for some index $j \in \{0, 1, \dots, N\}$. This results in exactly the same piecewise linear approximation as $\dots, t_{j-1}, t_j, \frac{t_j+t_{j+2}}{2}, t_{j+2}, \dots$ with the slopes $\dots, s_{j-1}, s_{j+1}, s_{j+1}, s_{j+2}, \dots$. In this way, the minimum should also be attained in the interior of \mathcal{T}_N .

3 Implementation issues and numerical results

Indeed, to find an optimal $\mathbf{t} \in \mathcal{T}_N$, we had to perform a conditional minimization. It turns out that unconditional minimization can harm the order of the components in \mathbf{t} .

To reduce the computational complexity, we introduce the following penalty term to avoid conditional minimization:

$$P_{\mathbf{t}} = K \cdot (|t_1 - 0| + |t_2 - t_1| + \dots + |1 - t_N| - 1),$$

where $K = 1000$ in the experiments. Clearly, if $\mathbf{t} \in \mathcal{T}_N$, this term should be zero. Altogether, we computed the minimum of

$$\mathbf{t} \rightarrow J_{\mathbf{s}(\mathbf{t}),\mathbf{t}} + P_{\mathbf{t}}$$

starting from a uniform division of the interval (a, b) . To approximate integrals in the loss term and in the finite element method, we applied a three-point Gauß integral. One can increase the accuracy of integration using the built-in Matlab subroutines but this does not increase further the accuracy of the final result. To compare our method with a similar one in [2], we use the same test problem

$$\begin{cases} \ddot{u}(x) = \frac{200}{9} \cdot \exp\{-100(x - \frac{1}{3})^2\} \cdot (1800 \cdot x^3 - 1200 \cdot x^2 + 173 \cdot x + 6) & x \in (0, 1) \\ u(0) = u(1) = 0, \end{cases} \quad (4)$$

where the analytic solution is given by $u(x) = x \cdot (\exp\{-100(x - \frac{1}{3})^2\} - \exp\{-400/9\})$.

The finite element solution, i.e. optimal piecewise linear approximation for (4) with the starting value \mathbf{t} and with the optimal \mathbf{t} are shown in Figure 1 and 2, respectively.

Also, we have tested the computational error of the adaptive finite element method given by the above optimization process in the $H_0^1(a, b)$ -norm. The results are shown in Table 3.

N	4	9	19	39	79
err _{ad}	4.45	0.637	0.305	0.155	0.0774
err _{un}	7.52	0.282	0.188	0.109	0.0608

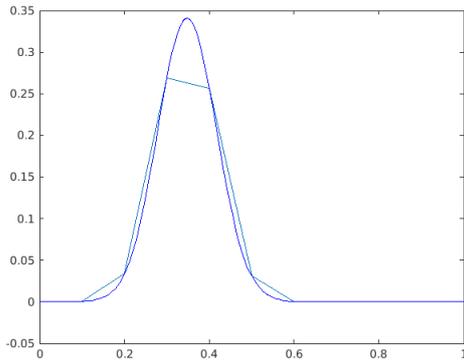


Figure 1: Finite element solution of (4) with $\mathbf{t} = (0.1, 0.2, \dots, 0.9)$ (dashed) together with the analytic solution of (4).

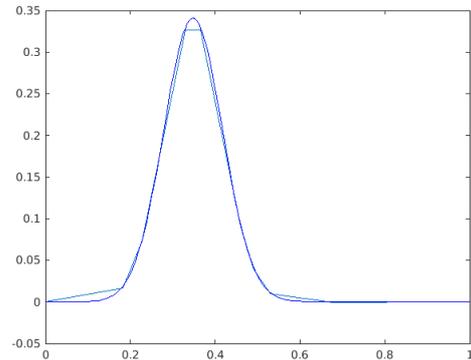


Figure 2: Finite element solution of (4) with the optimal \mathbf{t} together with the analytic solution of (4).

One can realize that the advance of adaptive methods is significant only in the case of relatively coarse meshes. On the other hand, the test problem in (4) has smooth solution. Therefore, on a sufficiently fine mesh, its solution can be approximated well also without adaptive refinement.

Acknowledgments

The research was supported by the Ministry of Innovation and Technology NRDI Office within the framework of the Artificial Intelligence National Laboratory Program.

References

- [1] **Axelsson, O., Barker, V.A.**, *Finite Element Solution of Boundary Value Problems*, Academic Press, San Diego, 1984.
- [2] **He, J., Li, L., Xu, J. and Zheng, C.**, ReLu deep neural networks and linear finite elements, *J. Comput. Math.*, **38**(3) (2020), 502–527.
- [3] **Kröse, B., van der Smagt, P.**, *An introduction to Neural Networks*, The University of Amsterdam, Amsterdam, 1996.
- [4] **Opschoor, J., Petersen, P. and C. Schwab, C.**, Deep ReLu networks and high-order finite element methods. *Anal. Appl.*, **18**(5) (2020), 715–770.

In search of an appropriate loss function for differential equations' initial value problems

Petra Csomós and Ferenc Izsák

ELTE Eötvös Loránd University, Institute of Mathematics

petra.csomos@ttk.elte.hu, ferenc.izsak@ttk.elte.hu

We consider the numerical treatment of ordinary differential equations' initial value problems where the approximate solution has the form of a two-layer neural network. Since the exact solution to the problem is unknown (and therefore there is no training set available), the parameters of the neural network (i.e., the approximate solution) originate from the minimisation of a loss function. Thus, the right choice of the loss function is inevitable. Namely, the appropriate loss function needs to perform well in numerical experiments, and should be minimal for the exact solution. In our work we present the properties of the neural network, and aim at deriving the form of the loss function by using the alternate forms of the error function.

1 Initial value problem

Let $T > 0$ and $d \in \mathbb{N}$ be arbitrary, $f: [0, T] \times \mathbb{C}^d \rightarrow \mathbb{C}^d$ be a continuous function, and $x_0 \in \mathbb{C}^d$ be a given vector. Then we consider the following initial value problem for the continuously differentiable unknown function $x: [0, T] \rightarrow \mathbb{C}^d$:

$$\begin{cases} x'(t) = f(t, x(t)), & t \in (0, T] \\ x(0) = x_0. \end{cases} \quad (1)$$

Our aim is to approximate the exact solution $x(t)$ for all $t \in [0, T]$ with the continuous function $x_n(t)$ being of the form of a neural network, where $n \in \mathbb{N}$ is arbitrary but fixed. We will show that $x_n(t)$ corresponds to a continuous piecewise linear function with n pieces of breaking points. The function $x_n(t)$ (via its parameters) should be then obtained by minimising an appropriate loss function $L(\Theta_n, T)$. Without loss of generality, we assume that $x_n(0) = x_0$.

2 Approximate solution as a neural network

Our aim is to approximate the exact solution x to problem (1) with $d = 1$ by the function $x_n: [0, T] \rightarrow \mathbb{C}$ given by the following two-layer neural network

$$x(t) \approx x_n(t) := \langle \mathbf{a}_2, \text{ReLu}(\mathbf{a}_1 t + \mathbf{b}_1) \rangle + b_2, \quad (2)$$

where $\langle \cdot, \cdot \rangle$ denotes the scalar product in \mathbb{C}^n , and the name ReLu stands for rectified linear unit, being a common activation function defined as

$$\text{ReLu}(y) := \max\{0, y\} = \frac{y + |y|}{2}, \quad y \in \mathbb{C}^n.$$

The parameters of the neural network are chosen as

$$\begin{aligned} \mathbf{a}_1 &:= (1, 1, \dots, 1) \in \mathbb{R}^n, & \mathbf{b}_1 &:= (0, -t_1, \dots, -t_{n-1}) \in \mathbb{R}^n \\ \mathbf{a}_2 &:= (s_1, s_2 - s_1, \dots, s_n - s_{n-1}) \in \mathbb{R}^n, & b_2 &:= x_0 \in \mathbb{C} \end{aligned}$$

with some $t_j > 0$, $s_j \in \mathbb{R}$ for all $j = 1, \dots, n$. We collect these parameters in the vector

$$\Theta_n := (x_0, t_1, \dots, t_{n-1}, s_1, \dots, s_n) \in \mathbb{C}^{2n}, \quad (3)$$

which depends on n and T . To ease the notation we do not indicate its dependency on T . Then the neural network $x_n(t)$ can be constructed as in (2) by using the parameter vector Θ_n defined in (3). The first important result is that the neural network $x_n(t)$ has exactly the piecewise linear form we expect.

Theorem 1 *For any $n \in \mathbb{N}$, the neural network $x_n: [0, T] \rightarrow \mathbb{C}$ of the form (2) with parameters (3) corresponds to a continuous piecewise linear function with $x_n(0) = x_0$, and has breaking points in $t_1 < t_2 < \dots < t_{n-1}$ and slopes s_j on (t_{j-1}, t_j) , $j = 1, \dots, n$. Moreover it has the recursive form*

$$x_n(t) = x_n(t_{j-1}) + (t - t_{j-1})s_j, \quad t \in [t_{j-1}, t_j], \quad j = 1, 2, \dots, n.$$

The proof relies on the direct reformulation of the representation (2).

Although, we presented the form (2) of the neural network in the case $d = 1$, one can define it in a similar manner also for any $d \in \mathbb{N}$ leading to a similar statement as in Theorem 1. Therefore, in the following, we will consider the general case.

3 Error functions

In numerical analysis, the convergence of the approximate solution x_n to the exact one x is shown by analysing the global error $\|x(T) - x_n(T)\|$. More precisely, the approximation is called convergent at $T > 0$ if $\|x(T) - x_n(T)\| \xrightarrow{n \rightarrow \infty} 0$. Then for any $0 \neq z_T \in \mathbb{C}^d$, the condition $\langle x(T) - x_n(T), z_T \rangle \xrightarrow{n \rightarrow \infty} 0$ is sufficient for the convergence. Thus, we will investigate the error function

$$\varepsilon_n(T) := \langle x(T) - x_n(T), z_T \rangle \quad \text{for any } z_T \neq 0, \quad (4)$$

and seek loss functions $L(\Theta_n, T)$ for which the condition $\|\varepsilon_n(T)\| \xrightarrow{n \rightarrow \infty} 0$ holds under some constraints. Due to the considerations above, this implies the convergence of the approximation, too. That is, if we find a suitable loss function, we ensure that the neural network possessing the parameters obtained by the minimisation of the loss function, will converge to the exact one for increasing number of breaking points (or time levels).

Hence, our next aim is to find a loss function with the property presented above. To do so, we consider the a posteriori error function proposed by Kehlet and Logg in [1]:

$$\widehat{\varepsilon}_n(T) = \int_0^T \langle (x'_n(t) - f(t, x_n(t))), z_n(t) \rangle dt, \quad (5)$$

where $z_n(t)$ is the solution to the adjoint equation which is defined as follows. For all $t \in [0, T]$, consider the linear operator (matrix)

$$A_n(t) := \int_0^1 (\partial_2 f)(t, sx_n(t) + (1-s)x(t)) ds \quad (6)$$

mapping $\mathbb{C}^d \rightarrow \mathbb{C}^d$. Then for any given $z_T \in \mathbb{C}^d$, the adjoint problem is defined as follows

$$\begin{cases} z_n'(\xi) = -A_n(\xi)^* z_n(\xi), & \xi: T \rightarrow 0 \\ z_n(T) = z_T, \end{cases} \quad (7)$$

where the star denotes the adjoint of the operator. We note that the variable ξ varies between T and 0 , that is, in the reverse order. Therefore, the adjoint problem (7) corresponds to the following one

$$\begin{cases} v_n'(t) = A_n(T-t)^* v_n(t), & t \in (0, T] \\ v_n(0) = z_T \end{cases} \quad (8)$$

for the unknown function $v_n: [0, T] \rightarrow \mathbb{C}^d$ defined as $v_n(t) := z_n(T-t)$ for all $t \in [0, T]$. This equivalence is used when analysing the error functions.

Our next step is to show that $\varepsilon_n = \widehat{\varepsilon}_n$ holds.

Theorem 2 *For any $T > 0$ and $0 \neq z_T \in \mathbb{C}^d$, consider problems (1) and (7) with solutions $x, z_n: [0, T] \rightarrow \mathbb{C}^d$, respectively. Let $x_n: [0, T] \rightarrow \mathbb{C}^d$ be of an arbitrary form. Then we have the relation $\varepsilon_n(T) = \widehat{\varepsilon}_n(T)$, that is,*

$$\langle x(T) - x_n(T), z_T \rangle = \int_0^T \langle (x_n'(t) - f(t, x_n(t))), z_n(t) \rangle dt$$

for any $T > 0$.

Proof: We present the idea of the proof.

As a first step we rewrite the original nonlinear problem (1) as the nonhomogeneous nonautonomous linear problem

$$\begin{cases} x'(t) = A_n(t)x(t) + g_n(t), & t \in (0, T] \\ x(0) = x_0 \end{cases} \quad (9)$$

with the linear operator (matrix) $A_n(t): \mathbb{C}^d \rightarrow \mathbb{C}^d$ already defined in (6) and

$$g_n(t) = f(t, x_n(t)) - A_n(t)x_n(t), \quad t \in [0, T].$$

Then by using the variation of constants formula, we have

$$x(T) = E_n(T, 0)x_0 + \int_0^T E_n(T, t)g_n(t) dt, \quad (10)$$

where $(E_n(t, s))_{0 \leq s \leq t \leq T}$ denotes the evolution family generated by the operator family $A_n(t)$. We note that for $d = 1$ we have

$$E_n(t, s) = \exp\left(\int_s^t A_n(\xi) d\xi\right), \quad 0 \leq s \leq t \leq T. \quad (11)$$

When rewriting the left-hand side of the statement, we use the form (10), the identities

$$\partial_t E_n(t, s) = A_n(t)E_n(t, s), \quad \partial_s E_n(t, s) = -E_n(t, s)A_n(s),$$

and the equivalence $v_n(t) = z_n(T - t)$, where v_n is the solution to (8). The latter implies, in particular, that $E_n(T, t)^* = F_n(T - t, 0)$, where $(F_n(t, s))_{0 \leq s \leq t \leq T}$ is the evolution family generated by $A_n(T - t)^*$. Putting all these together yields the assertion.

4 Loss functions

Loss functions can be constructed based on the error function in the proof of Theorem 2. For example, one can use a numerical approximation of the right-hand side of (11); we do not present the corresponding formulas here. It turns out that suitable approximations lead to loss functions $L(\Theta_n, T)$ with the property

$$\|\varepsilon_n(T)\| \leq L(\Theta_n, T). \tag{12}$$

We introduce the following notations for any $n \in \mathbb{N}$ fixed:

$$\begin{aligned} \Theta_n^* &:= \arg \min_{\Theta_n} L(\Theta_n, T), \\ x_n^*(T) &:= \text{the neural network (2) with parameters } \Theta_n^*, \\ \varepsilon_n^*(T) &:= x(T) - x_n^*(T). \end{aligned}$$

Now we can state our main result.

Theorem 3 *Suppose that property (12) holds. Then for any $T > 0$, we have the implication*

$$L(\Theta_n^*, T) \xrightarrow{n \rightarrow \infty} 0 \quad \Rightarrow \quad \|\varepsilon_n^*(T)\| \xrightarrow{n \rightarrow \infty} 0.$$

The proof is a direct consequence of property (12).

Acknowledgements

The research was supported by the Ministry of Innovation and Technology NRDI Office within the framework of the Artificial Intelligence National Laboratory Program.

References

- [1] **B. Kehlet and A. Logg**, A posteriori error estimation and global error control for ordinary differential equations by the adjoint method, *Numerical Algorithms*, **76** (2017), 191–210.

Solving the Laplace equation by using neural networks

Domonkos Haffner¹ and Ferenc Izsák²

Institute of Physics & AI ELTE Research Group, Eötvös Loránd University, Pázmány P. stny. 1.A, 1117 Budapest, Hungary ¹

Department of Applied Analysis and Computational Mathematics & MTA-ELTE, NumNet Research Group, Eötvös Loránd University, Pázmány P. stny. 1.C, 1117 Budapest, Hungary²

ferenc.izsak@ttk.elte.hu, haffner.domonkos@gmail.com

Abstract

A neural network approach is presented for solving Laplace equations. The mathematical basis of this approach is the boundary integral method. A shallow and simple neural network is used. The number of the parameters in the neural network is optimal, which results in a quick learning. A numerical experiment is also presented. We can avoid in this way the construction of grids or meshes, which is a significant advance of our approach.

1 Introduction

Neural networks are used nowadays in almost all area of scientific computing. In most cases, these are applied if no mathematical model was constructed to simulate or solve a real-life problem. Therefore, in most cases, large or even overly large parameter sets are applied, which can successfully fitted to the underlying problem. The main strength of this approach is the efficiency of this fitting, which can be considered also as an efficient optimization algorithm. For this, a wide arsenal of modern optimization methods are used including stochastic methods and automatic differentiation. Moreover, they are supported with program libraries and efficient subroutines.

We want to make use all of them to solve a basic but important benchmark problem in the numerical analysis. Laplace equations emerge frequently as a compound in real life problems by modeling rotation-free fluid dynamics, divergence-free electrodynamics or even simple diffusion, heat conduction processes [3].

For a detailed introduction of neural networks, we refer to [2]. From the mathematical point of view, we consider it as a function $\mathcal{NN} : \mathbb{R}^{d_0} \rightarrow \mathbb{R}^{d_N}$ depending on a parameter set Θ . In this framework, vectors in \mathbb{R}^{d_0} are called the input and vectors in \mathbb{R}^{d_N} the output vectors, respectively In this framework, vectors in \mathbb{R}^{d_0} are called the input and vectors in \mathbb{R}^{d_N} the output vectors, respectively In this framework, vectors in \mathbb{R}^{d_0} are called the input and vectors in \mathbb{R}^{d_N} the output vectors, respectively. Our task is to find the parameter set Θ_0 such that the corresponding function \mathcal{NN}_{Θ_0} approximates given input-output pairs $(\mathbf{u}_1, \mathbf{v}_1), (\mathbf{u}_2, \mathbf{v}_2), \dots, (\mathbf{u}_N, \mathbf{v}_N)$, as accurate as possible.

Formally, we have to solve the minimization problem

$$\min_{\Theta} \sum_{j=1}^N \rho(\mathcal{NN}_{\Theta}(\mathbf{u}_j), \mathbf{v}_j),$$

with some metrics ρ , or frequently, for some norm $\|\cdot\|$, we have the following:

$$\min_{\Theta} \sum_{j=1}^N \|\mathcal{N}\mathcal{N}_{\Theta}(\mathbf{u}_j) - \mathbf{v}_j\|^2.$$

2 Problem statement and methods

Recall, that the elliptic boundary value problem for the unknown function $u : \Omega \rightarrow \mathbb{R}$ with the Laplacian operator has the form

$$\begin{cases} \Delta u(\mathbf{x}) = 0 & \mathbf{x} \in \Omega \\ u(\mathbf{x}) = g(\mathbf{x}) & \mathbf{x} \in \partial\Omega, \end{cases} \quad (1)$$

where $\Omega \subset \mathbb{R}^2$ is a bounded Lipschitz domain and $g : \partial\Omega \rightarrow \mathbb{R}$ is given.

In real-life cases, however, the function g is known only in isolated points $\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_K$ of $\partial\Omega$ and also, the unknown function has to be determined in certain points $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_L$. Of course, in this way, we do not have a well-posed problem. Using continuous dependence of the solution on the boundary data, for a smooth function g , we obtain an accurate approximation of u , if it is computed based on the above discrete data.

To have training data, i.e. pairs $(\mathbf{u}_j, \mathbf{v}_j)_{j=1,2,\dots,N}$ above, we should know solutions in some cases. Here we use the concept of the fundamental solutions given with

$$\psi_{\mathbf{q}_j} : \mathbb{R}^2 \setminus \{\mathbf{q}_j\} \rightarrow \mathbb{R}, \quad \psi_{\mathbf{q}_j}(\mathbf{x}) = -\frac{1}{2\pi} \ln |\mathbf{x} - \mathbf{q}_j|.$$

We know that $\Delta\psi_{\mathbf{q}_j} = 0$ on $\mathbb{R}^2 \setminus \{\mathbf{q}_j\}$, and therefore, if $\mathbf{q}_j \notin \Omega$, this function delivers an input-output pair

$$\mathbf{u}_j = (\psi_{\mathbf{q}_j}(\mathbf{p}_1), \psi_{\mathbf{q}_j}(\mathbf{p}_2), \dots, \psi_{\mathbf{q}_j}(\mathbf{p}_K)) \quad \text{and} \quad \mathbf{v}_j = (\psi_{\mathbf{q}_j}(\mathbf{x}_1), \psi_{\mathbf{q}_j}(\mathbf{x}_2), \dots, \psi_{\mathbf{q}_j}(\mathbf{x}_L)).$$

Taking it for $j = 1, 2, \dots, N$, we obtain a number of N training data such that run a neural network to learn the solution the discrete Laplace equation.

The motivation of this approach is the so-called method of fundamental solutions introduced in [1]. Here the approximation u_h of u in (1) is sought in the following form:

$$u_h = \sum_{j=1}^N a_j \psi_{\mathbf{q}_j},$$

with some real coefficients $\{a_j\}_{j=1,2,\dots,N}$. Also, the points $\{\mathbf{q}_j\}_{j=1,2,\dots,N}$ are chosen to be outside of the domain in the near of the boundary. In this way, an appropriate linear combination of the boundary values $\{g(\mathbf{p}_k)\}_{k=1,2,\dots,K}$ will give all the values $u_h(\mathbf{x}_1), u_h(\mathbf{x}_2), \dots, u_h(\mathbf{x}_L)$.

Accordingly, we built a neural network with the following structure:

- Input size: K , output size: L .
- We had only one dense layer.
- We did not apply a bias.
- We did not apply any activation function.
- The initial weight $w_{k,l}$ was proportional with $(\|\mathbf{p}_k - \mathbf{x}_l\|$ for $k = 1, 2, \dots, K$ and $l = 1, 2, \dots, L$.

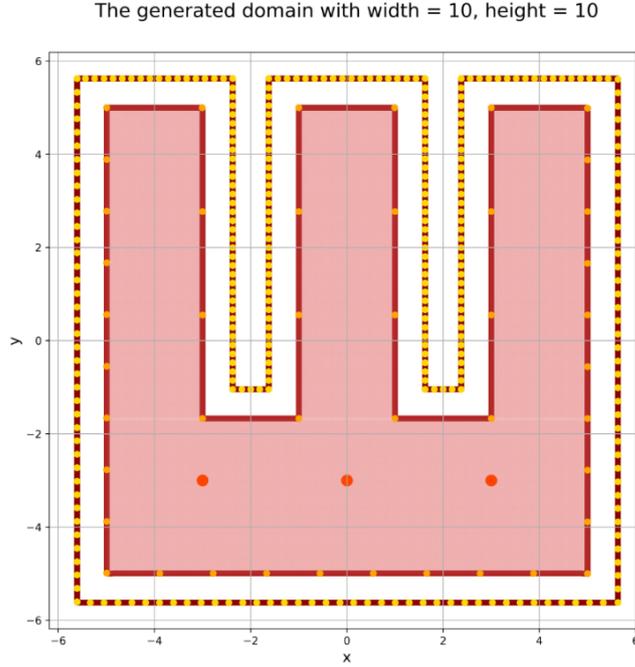


Figure 1: The computational domain with the boundary points \mathbf{p}_k , outer points \mathbf{q}_l and the three points x_1, x_2, x_3 in the domain.

3 Implementation issues and numerical results

To be convinced of the performance of our approach, we have chosen a non-trivial concave domain as shown in Figure 1.

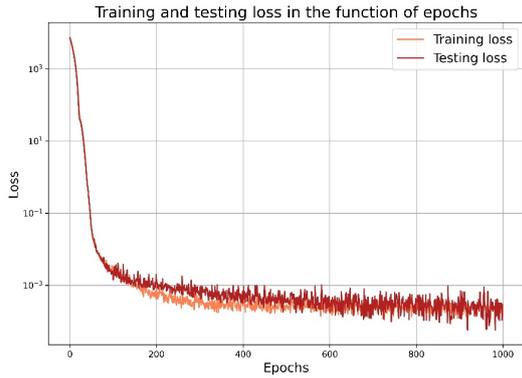


Figure 2: Training (orange) and testing (red) loss in a single run over 1000 epochs.

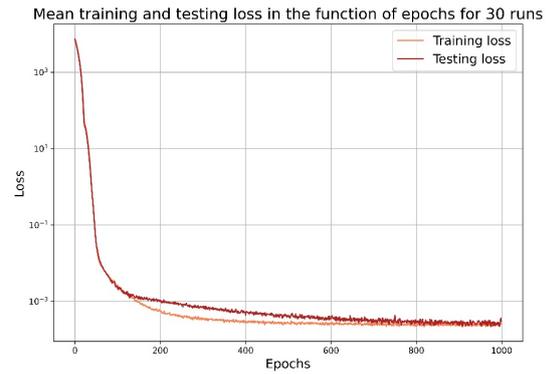


Figure 3: Training and testing loss of an average of 30 independent runs over 1000 epochs.

The neural network was trained over 1000 epochs. On a simple laptop, this ran approximately over 35 seconds. We applied 0.03 as the learning rate, the size K of the input was 52, while the size K of the input was 3. We have trained the network with 288 data

pairs and validated the results using 32 pairs.

The method was implemented in Python using the Keras deep learning library.

To smooth the results, we took the average of 30 independent runs. The corresponding errors are shown in Figure 2 and 3, respectively. In all cases, we have measured the average squared error loss in the three points \mathbf{x}_1 , \mathbf{x}_2 and \mathbf{x}_3 . One can realize that training and test losses are decreasing with a very similar rate. This is also an indicator to have an efficient neural network.

Until now, the test cases covered only the singular functions $\psi_{\mathbf{q}_j}$. To complete this, Therefore, we have also tested our network for the given solution $u(x, y) = (x+5)^2 - (y-2)^2$ of (1) using the above domain. In all cases, we obtained an accurate prediction and after training, the computational time to solve the problem was only 1 ms.

Summarized, this approach seems to be very efficient and simple most importantly, because we do not need to generate a grid or mesh for the numerical solution.

Acknowledgments

The Project is supported by the Hungarian Government and co-financed by the European Social Fund; EFOP-3.6.3-VEKOP-16-2017-00001: Talent Management in Autonomous Vehicle Control Technologies.

The research was also supported by the Ministry of Innovation and Technology NRDI Office within the framework of the Artificial Intelligence National Laboratory Program.

References

- [1] **Mathon, R., Johnston, R. L.**, The approximate solution of elliptic boundary-value problems by fundamental solutions. *SIAM J. Numer. Anal.* **14**(4) (1977), 638–650.
- [2] **Kröse, B., van der Smagt, P.**, *An introduction to Neural Networks*, The University of Amsterdam, Amsterdam, 1996.
- [3] **Rappaz, M., Bellet, M., Deville, M.** *Numerical Modelling in Materials Science and Engineering*, Springer, Berlin, Heidelberg, 2003.

Residual neural networks as numerical approximations of differential equations

Gábor Hidy

Eötvös Loránd University
Institute of Mathematics
AI Research Group
hidygabor@student.elte.hu

Abstract

Residual neural networks – ResNets, for short – are a powerful form of convolutional neural networks, widely used in image recognition tasks. Their main building block, the residual block, is of the same form as the update rule for a forward Euler scheme, therefore the network itself can be thought of as an approximation of the solution of a differential equation.

There are other numerical methods for approximating such a function, including the linear multistep method. A modified ResNet architecture, christened the LM-ResNet, has been proposed with a structure based on the update rule of said scheme. The original article introducing it claimed that the network had shown an improved performance with standard image classification tasks. I attempt to reproduce some of these experiments and verify the usefulness of the LM architecture.

1 Introduction

Convolutional neural networks (CNNs) have been the leading tools in artificial intelligence based computer vision for a decade now. Their structures require very few per-layer parameters, which allows for more than a dozen layers in one network, a depth that is hardly achievable without the convolutional structure.

However, plain CNNs with significantly more than 20 layers are still not practical, since their performance seems to worsen after this threshold. In recent years, residual neural networks have been proposed as a possible solution for this problem. [2]

The ResNet networks proposed in [2] and [3] consist of residual blocks of shape

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{f}(\Theta, \mathbf{x}_k) \quad (1)$$

where \mathbf{f} represents two (or, in some cases, three) consecutive convolutional layers. This architecture allows for networks with up to a hundred, or, in some cases, more than a thousand layers, that achieve record-breaking performances on standard computer vision tasks.

ResNets have a natural connections with solutions of first-order differential equations. The key observation concerning the link between the two is that equation (1) can be rewritten, with $\Delta t = 1$, as $\mathbf{x}_{k+1} = \mathbf{x}_k + \Delta t \mathbf{f}(\Theta_k, \mathbf{x}_k)$, which is a step of a forward Euler scheme approximating the solution $\mathbf{x}(t)$ to the differential equation

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\Theta(t), \mathbf{x}(t)) \quad (2)$$

Remark 1 *Instead of $\Delta t = 1$, equation (1) can be written as $\mathbf{x}_{k+1} = \mathbf{x}_k + \frac{1}{n}\mathbf{f}(\Theta'_k, \mathbf{x}_k)$, since all components of \mathbf{f} – e. g. the convolutional, batch normalisation and ReLU layers – are positively homogenous. This form is supported by the fact that in deep residual networks, the residual functions $\mathbf{f}(\Theta_k, \cdot)$ are close to zero. [1] This reinforces the notion that deeper ResNets simply correspond to an approximation of equation (2) with more iterations.*

2 The LM architecture

The LM-ResNet model, proposed by Lu et al. [5], is a modification of the plain ResNet architecture, motivated by its connection with differential equations. The LM modification of a ResNet architecture replaces residual blocks seen in equation 1 with LM residual blocks

$$\mathbf{x}_{k+1} = \vartheta_k \mathbf{x}_{k-1} + (1 - \vartheta_k) \mathbf{x}_k + \mathbf{f}(\mathbf{x}_k) \quad (3)$$

While a plain residual block can be associated with the forward Euler scheme, the above blocks are motivated by the linear multistep method (hence the name LM-ResNet) approximation of the same differential equation, described in equation (2). The parameters ϑ_k are trainable, and at initialisation sampled from a uniform distribution on $[-\frac{1}{10}, 0]$.

So the construction of an LM-ResNet is as follows: take a plain ResNet – more precisely, a ResNet using preactivational residual units, colloquially known as ResNetV2 [3] – and follow every residual block with the LM block in equation (3). This adds a few extra parameters, but the change is negligible – ResNet-20 has more than 270 000 parameters, and only 9 residual blocks. It is important to note however, that the above construction is lacking. It does not account for two problems: changing dimensions, and the calculation of \mathbf{x}_1 .

A typical ResNet architecture consists of three stacks of layers, each of which has different output dimensions. The plain residual blocks of equation 1 are not sufficient at the border of these stacks, so parameterless padding shortcuts are used instead, where the extra dimensions are padded with 0s. This construction is also sufficient for LM-ResNet.

Conversely, the second problem does not appear in a ResNet architecture, which uses a one-step method. Generally, a (linear) m step method only works if the first m steps have been calculated by a different method. Following this practice, the first block can be a simple residual block (option A).

An alternative solution would be to, instead of an identity shortcut, apply a projections shortcut, so

$$\mathbf{x}_1 = \mathbf{C}\mathbf{x}_0 + \mathbf{f}(\mathbf{x}_0) \quad (4)$$

where \mathbf{C} is a 1×1 convolution operation (option B). This again increases the number of parameters in the network, but only by 272 parameters, which is, in all cases, less than 0.1% of all parameters.

Remark 2 *Lu et al.’s solutions to the above problems are not made clear in [5], which is why I have made my own assumptions.*

Model	Residual unit	Error		
		Mean	Best	Reported
ResNet-20	postactivational	8.34%	8.19%	8.75%
ResNet-32	postactivational	7.67%	7.29%	7.51%
ResNet-44	postactivational	7.94%	7.16%	7.17%
ResNet-56	postactivational	7.60%	6.89%	6.97%
ResNet-110	postactivational	7.19%	6.79%	6.61%
ResNet-164	postactivational	6.28%	5.90%	5.93%
ResNet-20	preactivational	8.58%	8.12%	–
ResNet-32	preactivational	7.64%	7.51%	–
ResNet-44	preactivational	7.09%	6.98%	–
ResNet-56	preactivational	6.76%	6.61%	–
ResNet-110	preactivational	6.28%	6.06%	6.37%
ResNet-164	preactivational	5.52%	5.18%	5.46%
LM-ResNet-20 A	preactivational	8.62%	8.37%	8.33%
LM-ResNet-20 B	preactivational	8.52%	8.45%	
LM-ResNet-32 A	preactivational	7.77%	7.60%	7.18%
LM-ResNet-32 B	preactivational	7.63%	7.28%	
LM-ResNet-44 A	preactivational	7.10%	6.89%	6.66%
LM-ResNet-44 B	preactivational	7.02%	6.84%	
LM-ResNet-56 A	preactivational	6.91%	6.65%	6.31%
LM-ResNet-56 B	preactivational	6.58%	6.28%	
LM-ResNet-110 A	preactivational	6.07%	5.80%	–
LM-ResNet-110 B	preactivational	6.17%	5.98%	

Table 1: Different models’ mean and best test error, compared to the error reported in the original articles

3 Results

Lu et al. have conducted several experiments with the LM-ResNet networks, and concluded that compared to their plain counterparts, LM architectures of every depth fared better. I argue that, while their results look promising, some of the comparisons they make are not adequate.

My focus will be on their results measured on the CIFAR-10 dataset. [4] Lu et al. tested architectures of depths 20, 32, 44, and 56 layers on CIFAR-10. They compared

their results to the performance of ResNets of the same depths published in [2], and to that of the ResNet-110, published in [3]. However, all of the original results were measured with postactivational units, while Lu et al. uses preactivational units all over. This weakens the claim that the LM architectures improves performance, since preactivational units have been known to perform better, albeit they are usually employed in deeper networks.

The other, main difference between Lu et al.’s and the original experiments is the size of the training set. The CIFAR-10 dataset consists of 60 000 images, 10 000 of which are traditionally used as a testing set. Lu et al. used all the remaining 50 000 images as training datapoints, while the original ResNet results were measured on models trained on a 45 000 image training set, with 5000 images used as validation.

I have attempted to recreate these experiments, as well as some that were missing from the original papers. Hyperparameters and other details of the training matched those in [2]. Table 1 shows the results. Each experiment was run five times to account for random initialisation. This is contrary to the results reported in [2], where only the ResNet-110 tests were run multiple times. This is probably responsible for the fact that the reported error usually lies between my best and mean errors, in the case of ResNets. This is not the case for LM-ResNets, which is at least partially due to my using the smaller, 45 000 picture training set.

The experiments show that, although not by a large margin, LM-ResNets do perform better, even when compared against ResNets using preactivational units. They also confirm that preactivational units are a better building block for ResNets, although their effect is less significant in shallower models. Experiments run with LM-ResNet-110 – that were missing from [5] – show that switching to the LM architecture also affects deep models more, which is another evidence to the claim that the better performance of the LM-ResNets is due to the change in architecture, not the few extra parameters.

References

- [1] **De, S. and Smith, S. L.**, Batch normalization biases residual blocks towards the identity function in deep networks, in: H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan and H. Lin (Ed.) *NeurIPS 2020* (virtual, 2020), Advances in Neural Information Processing Systems **33**, Curran Associates, 2020, 19 964–19 975.
- [2] **He, K., Zhang, X., Ren, S., and Sun, J.**, Deep residual learning for image recognition, in: *CVPR 2016* (Las Vegas, 2016), Conference on Computer Vision and Pattern Recognition, IEEE, 2016, 770–778.
- [3] **He, K., Zhang, X., Ren, S., and Sun, J.**, Identity mappings in deep residual networks, in: B. Leibe, J. Matas, and M. Welling (Ed.) *Computer Vision – ECCV 2016* (Amsterdam, 2016), European Conference on Computer Vision, Springer, 2016, 630–645.
- [4] **Krizhevsky, A.**, Learning Multiple Layers of Features from Tiny Images, <https://www.cs.toronto.edu/%7Ekriz/learning-features-2009-TR.pdf>
- [5] **Lu, Y., Zhong, A., Li, Q., and Dong, B.**, Beyond finite layer neural networks: Bridging deep architectures and numerical differential equations, in: J. Dy and A. Krause (Ed.) *Proceedings of the 35th International Conference on Machine Learning* (Stockholmsmässan, 2018), Proceedings of Machine Learning Research **80**, PMLR, 2018, 3276–3285.

Learning a function from data by solving a differential equation and tuning its parameters

András Molnár, Imre Fekete and Péter L. Simon

AI Research Group and Institute of Mathematics, ELTE TTK
`{andras.molnar, imre.fekete, peter.simon}@ttk.elte.hu`

1 Introduction

In the recent years, machine learning has been connected to the field of differential equations by people finding formulae reminiscent of that of numerical time integrators inside neural networks [3, 2]. This has led to the discovery of the adjoint method as a continuous analogue of backpropagation [1]. In this manuscript, inspired by [1], we consider the problem of finding a differential equation such that the trajectories of some of its solutions best fit a set of data.

We start with the mathematical formulation of the problem. Given a time interval $[t_1, t_N]$ with one of its subsets S containing t_1 , a sample of time-value pairs $\{(s, g(s))\}_{s \in S} \subset (\mathbb{R} \times \mathbb{R}^n)^S$, a set of parameters Θ , and a family of functions locally Lipschitz in their second variable

$$\{f_\theta : \mathbb{R} \times \mathbb{R}^n \rightarrow \mathbb{R}^n \mid \theta \in \Theta\},$$

we consider the family of initial value problems

$$\begin{cases} \dot{x}(t) = f_\theta(t, x(t)) & t_1 < t < t_N, \theta \in \Theta, \\ x(t_1) = g(t_1), \end{cases}$$

and seek a parameter $\theta \in \Theta$ such that the corresponding solution x_θ provides a sufficiently good approximation of the sample in the sense that the functions $x_\theta|_S$ and g are sufficiently close in some seminorm.

We refer to this process as learning (via) a differential equation. Either theoretical, when S is an interval; or practical, when it is a discrete set. The latter case is the one that allows for numerical experiments, which we carry out using the adjoint method outlined in [1].

As we shall soon see, the empirical data is sometimes better described with a differential equation, the dimension of which is different than that of the data. In this case, first, the data may be lifted, then the solution of the equation projected back to the original dimension. To achieve this, we will use a simple linear transformation P and its transpose P^T . We prefer not to use more sophisticated mappings, especially ones which depend on the parameter θ , since doing so may obscure the learning ability of the differential equation.

In the following pages we will mainly restrict our attention to families of linear autonomous systems, that is, we consider families of the form

$$\begin{cases} \dot{x}(t) = Ax(t) + b & t_1 < t < t_N, \\ x(t_1) = P^T g(t_1). \end{cases}$$

We shall seek to choose the unknown parameter $\theta = (A, b) \in \mathbb{R}^{n \times n} \times \mathbb{R}^n$ such that an L^p seminorm of the differences is minimized. In our numerical experiments, we start from a discrete sample with the set of times $S = \{t_1 \dots t_N\}$. Thus, we choose to minimize the p th power of the discrete L^p norm via the following loss function

$$L(\theta) = \sum_{j=2}^N (t_j - t_{j-1}) |Px_\theta(t_j) - g(t_j)|^p,$$

where P is the aforementioned linear transformation. Unless stated otherwise, P simply discards some of the coordinates, and $p = 1$.

2 Investigating simple physics problems

In this section we consider some physics problems and show that they can be learnt both in theory and in practice.

2.1 Free fall

As a motivating example, consider the perfect free fall of a body. Can we learn what differential equation is governing its movement from a set of measurements of its position? The model says that if the position and speed of the body at time 0 are 0, then the position of the body at time $0 \leq t \leq 1$ is going to be

$$g(t) = \frac{10}{2}t^2.$$

The natural model leads to the two dimensional system

$$\frac{d}{dt} \begin{bmatrix} x \\ \dot{x} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \end{bmatrix} + \begin{bmatrix} 0 \\ 10 \end{bmatrix} \quad (1)$$

considered on $(0, 1)$; with initial value $P^T g(0) = [1 \ 0]^T \cdot g(0) = [g(0) \ 0]^T$.

This means that the theoretical learning of g is possible with a family of two dimensional autonomous linear system that contains this matrix vector pair.

Considering the practical learning aspect, we can say that starting from a small enough neighbourhood of optimal parameters, the error term is small, and it decreases as the optimization proceeds. If the above sparsity pattern is enforced, then the learning proceeds more confidently, as witnessed by Figure 1.

Lastly, we remark that some further examples of families of differential equations that permit theoretical learning in this case are

$$f_{\theta_1}(t, x) = \theta_1 \sqrt{x}, \quad \text{or} \quad f_{\theta}(t, x) = \theta_2 x + \theta_3 t + \theta_4,$$

with optimal parameters $\theta_1 = 2\sqrt{5}$, and $(\theta_2, \theta_3, \theta_4) = (0, 10, 0)$ respectively. Interestingly, the physically correct one is higher dimensional.

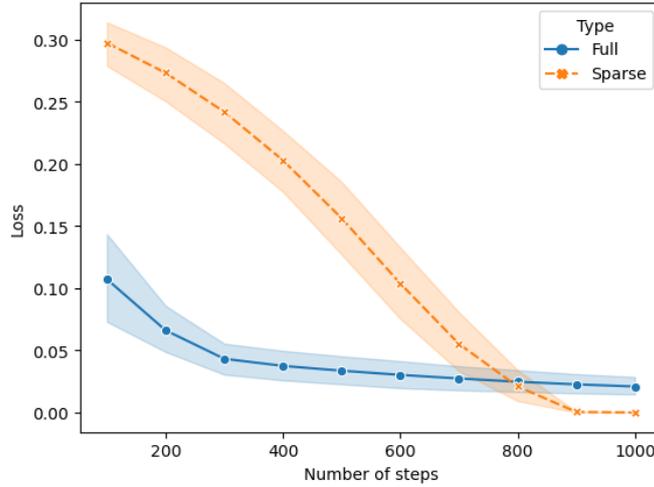


Figure 1: Loss values encountered while learning t^2 on the unit interval with and without enforcing the sparsity pattern seen in (1). The 20-20 initial matrix vector pairs have been randomly selected with $[0,1]$ -uniformly distributed coordinates.

2.2 Harmonic oscillator

Another simple example is the harmonic oscillator, that allows us to learn the cosine function. A two dimensional homogeneous autonomous linear family containing matrices that are square roots of the negative identity provides a natural choice.

More precisely, learning this function on the unit interval is possible using the transformation $P = \begin{bmatrix} 1 & 0 \end{bmatrix}$, if the family includes the matrix

$$A = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}.$$

We remark that these are not the only possible choices. Indeed, on the same interval, we could also conjugate this matrix with an orthogonal matrix Q and replace the previous P with the transformation PQ .

In practice, with initial matrix A , some apriori steps to minimize $\|A^2 + I\|_F$, and adding this term to the loss function speeds up learning, as seen in Figure 2.

3 Predicting the daily mean temperature

So far we have considered problems where exact theoretical learning was possible, since we have selected the function g and the family $\{f_\theta\}_{\theta \in \Theta}$ in a way that for some known $\theta \in \Theta$, the equality $Px_\theta|_{\mathcal{S}} = g$ could be achieved. In this section we apply the methodology to predict the daily mean temperature in Budapest, a problem, where an optimal right hand side f_θ is not known to us.

The prediction process is as follows. Given 30 successive daily temperature measurements, we pair them with time points t_j from the unit interval, endpoints included, with uniform spacing between them. We attempt to approximate these data points using the 2

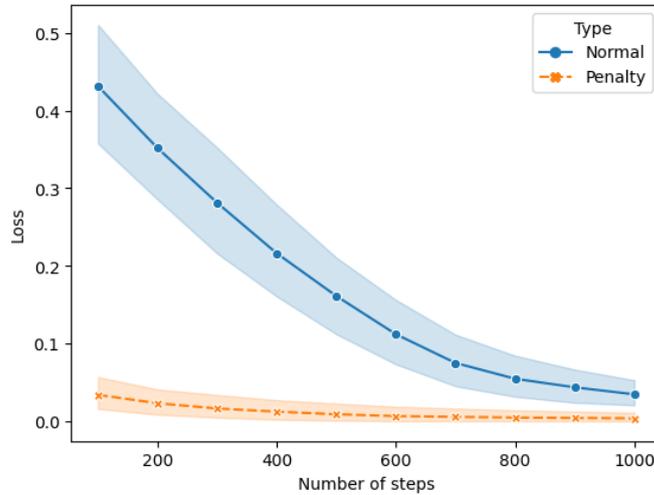


Figure 2: Loss values encountered while learning \cos on the unit interval with and without the initialization and penalty term $\|A^2 + I\|_F$. The 20-20 initial matrices have been randomly selected with $[0,1]$ -uniformly distributed coordinates.

dimensional linear autonomous family, and $P = \begin{bmatrix} 1 & 0 \end{bmatrix}$. When a sufficiently low loss value is reached, we solve the initial value problem to time $1 + \frac{1}{29}$, which yields our prediction of the temperature of the 31st day.

So far, the prediction power of this model seems limited. Perhaps this is explained by our insistence on purity. While the choice of P could be reconsidered, finding a family, such that its solutions are dense in some suitable function space, and therefore approximate our data well, seems the most promising direction.

4 Acknowledgements

The project has been supported by the European Union, co-financed by the European Social Fund (EFOP-3.6.3-VEKOP-16-2017-00002). The research was supported by the Ministry of Innovation and Technology NRDI Office within the framework of the Artificial Intelligence National Laboratory Program.

References

- [1] **R. T. Chen, Y. Rubanova, J. Bettencourt, D. Duvenaud**, Neural ordinary differential equations, *Advances in Neural Information Processing Systems 31 (NeurIPS 2018)*, pp. 6571-6583, 2018
- [2] **Y. Lu, A. Zhong, Q. Li, B. Dong**, Beyond Finite Layer Neural Networks: Bridging Deep Architectures and Numerical Differential Equations, *Proceedings of the 35th International Conference on Machine Learning, PMLR 80:3276-3285*, 2018
- [3] **L. Ruthotto, E. Haber**, Deep Neural Networks Motivated by Partial Differential Equations, *Journal of Mathematical Imaging and Vision, Vol. 62*, pp. 352–364, 2020

Saddle-node bifurcation in a 3-dimensional neural network model

Anita Windisch*

Department of Applied Analysis and Computational Mathematics, Eötvös Loránd
University
gyuricsek@caesar.elte.hu

1 Introduction

Recurrent neural networks are widely used for both industrial and scientific purposes such as in speech and manuscript recognition or to generate subtitles automatically. These networks can be represented as a directed graph where the nodes are the neurons and every edge has a weight which means the strength of the connection. In a recurrent neural network the signals coming out from a neuron can flow back to itself via other nodes so the spread of the information is not obvious in contrast to the feedforward networks.

In this work the Hopfield model is investigated which can be applied as autoassociative memory and it has the ability to retrieve the stored data from partial information. The Hopfield model is a system of ordinary differential equation which takes the form [3]

$$\dot{x} = -Dx + Wy + I, \quad y_i = f(x_i), \quad (1)$$

where x is the membrane potential and y is the firing rate of the neurons, the matrix W contains the strengths of connections, I is the external input and f is the activation function. The main goal is to determine those parameter values for which the number of equilibria changes, i.e. the saddle-node bifurcation curve when the weight matrix has a special structure. The fixed points in this case can be considered as memory states of the network. To investigate the dynamics of the Hopfield model we use analytical tools and MATCONT [2] which is a numerical toolbox of MATLAB for the study of parametrized dynamical systems and we apply the sigmoid function $f(x) = (1 + \exp(a - bx))^{-1}$ as activation function.

Different variants of the Hopfield model have already been investigated from several aspects. R. D. Beer approximated the bifurcation curves by hyperplanes and determined the probability of choosing parameter values from a given domain in the parameter space ϑ_i [1]. D. Fasoli, A. Cattani and S. Panzeri partitioned the neurons according to the sign of their weights. The dynamics of the model was described in their work depending on the external input using numerical tools [4]. Despite the fact that the model has already been investigated in many cases there are still several to be studied. We consider a fully connected network with the following assumptions. Let the signals coming from a given

***Acknowledgement** Supported by the project "Integrated program for training new generation of researchers in the disciplinary fields of computer science", No. EFOP-3.6.3-VEKOP-16-2017-00002. The project has been supported by the European Union and co-funded by the European Social Fund.

neuron have the same weight and assume that none of the neurons are connected to themselves directly. Then the weight matrix takes the form

$$W = \begin{pmatrix} 0 & w_2 & \dots & w_n \\ w_1 & 0 & \dots & w_n \\ \vdots & \vdots & \ddots & \vdots \\ w_1 & w_2 & \dots & 0 \end{pmatrix}.$$

Let us further assume that the neurons do not receive any external input and let the matrix D be the identity. After these assumptions the Hopfield model (1) can be written as

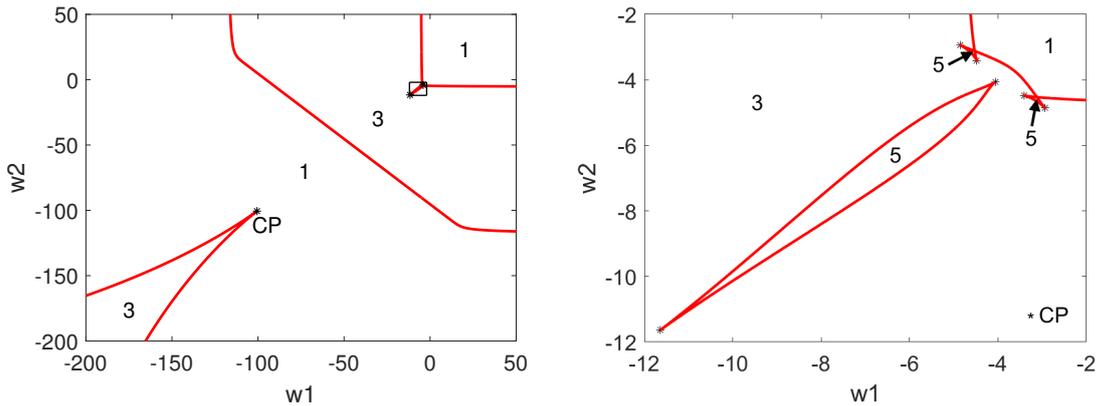
$$\dot{x}_i = -x_i + \sum_{k=1}^n w_k f(x_k) - w_i f(x_i). \quad (2)$$

2 The 3-dimensional model

Now let us consider a network which contains a single neuron with weight $w_1 \in \mathbb{R}$, another neuron with weight $w_2 \in \mathbb{R}$ and $n - 2$ neurons with the same positive weight w . Then Theorem 1 in paper [5] can be applied to reduce model (2) to a lower dimensional system. Therefore system

$$\begin{aligned} \dot{x}_1 &= -x_1 + w_2 f(x_2) + (n - 2)w_3 f(x_3) \\ \dot{x}_2 &= -x_2 + w_1 f(x_1) + (n - 2)w_3 f(x_3) \\ \dot{x}_3 &= -x_3 + w_1 f(x_1) + w_2 f(x_2) + (n - 3)w_3 f(x_3) \end{aligned} \quad (3)$$

determines the asymptotic behaviour of the considered network. The weights w_1 and w_2 are chosen as bifurcation parameters while the other parameters are fixed. Figure 1 shows



(a) Larger domains of the parameter plane. The rectangle shows the magnified part on the right. (b) Swallowtails and the island are magnified.

Figure 1: Saddle-node bifurcation curves and the number of steady states in system (3) with $w = 15$, $a = 4$, $b = 1$ and $n = 10$. The black stars denote the cusp points (CP).

the saddle-node bifurcation curves in system (3) detected by MATCONT when $w = 15$, $a = 4$, $b = 1$ and $n = 10$. As it can be seen the model can have 1, 3 or 5 steady states and the bifurcation curves form 2 swallowtails and an island which are magnified in Figure 1b.

If the value of weight w is changed to 80 then a more complex bifurcation diagram can be observed which is shown in Figure 2. In this case such domains are also found on the parameter plane where the system can have 7 or 9 steady states besides those ones where there are 1, 3 or 5 equilibria.

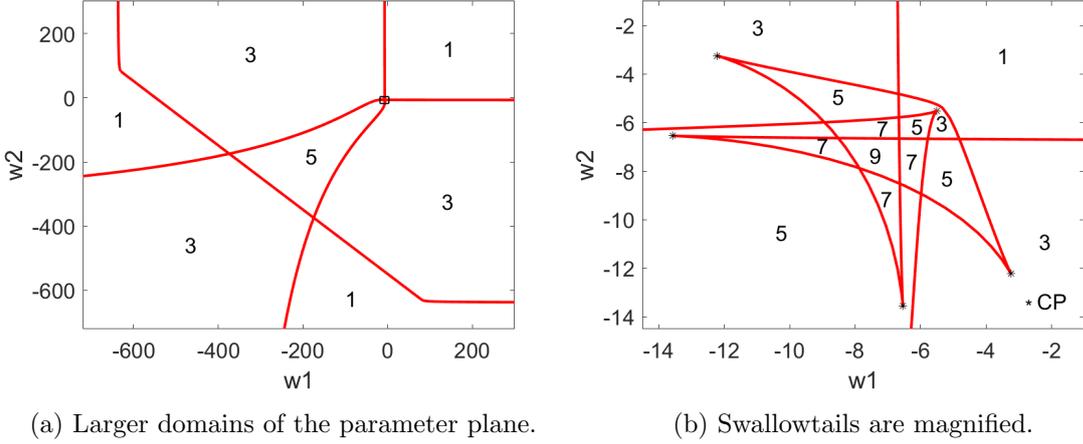


Figure 2: Saddle-node bifurcation curves and the number of steady states in system (3) with $w = 80$, $a = 4$, $b = 1$ and $n = 10$. The black stars denote the cusp points (CP).

3 The n -dimensional model

In the following our goal is to give a general approach to formalize the saddle-node bifurcation in a network which contains n neurons with different and arbitrary weights. In this case the Hopfield model takes the form of equation (2). First of all let us determine the steady states by putting the right hand side of the system equal to zero. If we introduce $q := \sum_{j=1}^n w_j f(x_j)$ then equation

$$x_i + w_i f(x_i) = q \quad \forall i = 1, \dots, n \quad (4)$$

determines the equilibria. After that let us take the solution of equation (4) and denote it by $\varphi(q, w)$. We note that equation (4) can have several solutions so φ is not unique for all q and w . If φ is substituted into the formula of q then equation

$$F(q) = 0, \quad \text{where} \quad F(q) := q - \sum_{j=1}^n w_j f(\varphi(q, w_j)) \quad (5)$$

determines the steady states.

To get the saddle-node bifurcation curve let us take the derivative of equation (5) which determines when the number of the solutions changes in equation (4). Then $F'(q) = 0$ takes the form $1 - \sum_{j=1}^n w_j f'(\varphi(q, w_j)) \partial_q \varphi(q, w_j) = 0$.

After that if we substitute φ into equation (4) and differentiate it with respect to q we get $\partial_q \varphi + w f'(\varphi) \partial_q \varphi = 1$, from which $\partial_q \varphi$ can be expressed. After some algebra we get that equations

$$q = \sum_{j=1}^n w_j f(\varphi(q, w_j)), \quad n - 1 = \sum_{j=1}^n \frac{1}{1 + f'(\varphi(q, w_j))}$$

determine the saddle-node bifurcation in system (2).

This analytical formalization were implemented in MATLAB for the 3-dimensional case detailed in Section 2. Results were plotted by blue stars in Figure 3 applying $w = 80$ while the red curves are detected using MATCONT. As it can be seen, the results of the analytic and numerical method fit together.

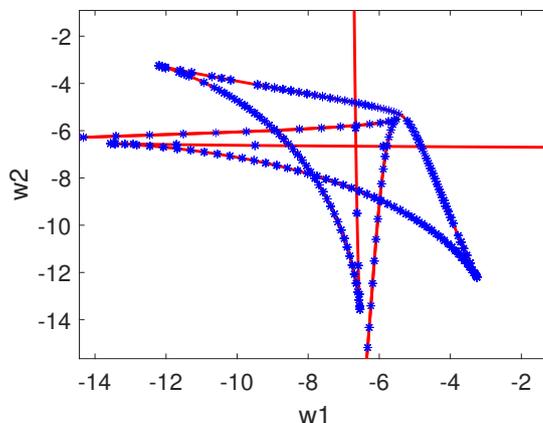


Figure 3: Saddle-node bifurcation in system (3). Red curves are calculated numerically, blue stars are the analytic results.

References

- [1] **Beer, R. D.**, Parameter Space Structure of Continuous-Time Recurrent Neural Networks. *Neural Computation* **18**(12), 3009-3051, (2006).
- [2] **Dhooge, A., Govaerts, W., Kuznetsov, Y. A.**, MATCONT: a MATLAB package for numerical bifurcation analysis of ODEs. *ACM Transactions on Mathematical Software (TOMS)* **29**(2), 141-164, (2003).
- [3] **Ermentrout, B., Terman, D. H.**, *Mathematical foundations of neuroscience* (Vol. 35). Springer Science & Business Media (2010).
- [4] **Fasoli, D., Cattani, A., Panzeri, S.**, The complexity of dynamics in small neural circuits. *PLoS computational biology* **12**(8), e1004992, (2016).
- [5] **Windisch, A., Simon, P.L.**, The dynamics of the Hopfield model for homogeneous weight matrix, *Ann. Univ. Sci. Budapest. Sect. Math* (accepted)

Section:

Numerical solution of differential equations, qualitative properties and applications

Organizer: István Faragó

Invited talk:

Róbert Horváth: Numerical solution of differential equations, qualitative properties and applications

Contributions:

- Teshome Bayleyegn and Ágnes Havasi: The method of multiple Richardson extrapolation
- Lívia Boda: Operator splitting and Average method
- Gabriella Svantnérné Sebestyén: Application of the Carleman linearisation method to partial differential equations
- Bálint Takács, Róbert Horváth István Faragó and Yiannis Hadjimichael: Numerical methods for space-dependent epidemic models



Numerical solution of differential equations, qualitative properties and applications

Róbert Horváth

Department of Analysis
Budapest University of Technology and Economics and
MTA-ELTE Numerical Analysis and Large Networks Research Group
Egry J. u. 1, H-1111 Budapest, Hungary
rhorvath@math.bme.hu

1 Introduction

A lot of problems in applications can be written in mathematical form as a differential equation. Because the original problems generally have some characteristic properties, it is natural to require the analogous versions of these properties for the numerical solution. In this short paper, we consider the decrease of the local extremizers of certain nonlinear parabolic problems and give conditions that guarantee the same property for the numerical solution obtained with the explicit Euler method.

2 Decrease of the number of the local extremizers of a special nonlinear parabolic problem

Let T be a positive real number and define the sets $Q_T = (0, T) \times (0, 1)$, $\bar{Q}_T = [0, T] \times [0, 1]$, $\Gamma_0 = \{0\} \times [0, 1]$ (initial time boundary of \bar{Q}_T), $\Gamma_T = \{T\} \times [0, 1]$ (final time boundary of \bar{Q}_T). Let us consider the nonlinear problem

$$\begin{aligned} \partial_t u &= r(t, x, u, \partial_x u, \partial_x^2 u), \quad (t, x) \in Q_T, \\ u(0, x) &= u_0(x), \quad x \in (0, 1), \\ u(t, 0) &= u(t, 1) = 0, \quad t \in [0, T], \end{aligned} \tag{1}$$

where we assume that the right-hand side function $r : \bar{Q}_T \times \mathbb{R}^3 \rightarrow \mathbb{R}$, $(t, x, r_3, r_4, r_5) \mapsto r(t, x, r_3, r_4, r_5)$ and the initial function $u_0 : (0, 1) \rightarrow \mathbb{R}$ are sufficiently smooth and guarantee the existence and the uniqueness of the solution $u \in C^{1,2}(\bar{Q}_T)$.

We recall Nickel's paper that proves the fact that the number of the local extremizers decreases. An interval (or specially a point) $I \subset [0, 1]$ is a local maximizer of the function $x \mapsto u(t, x)$, $x \in [0, 1]$ if the function value is constant in I , the interval I cannot be extended with this property, but the interval can be extended in both directions (if $I = [0, 1]$ then this condition can be omitted, if only $0 \in I$ or $1 \in I$ then we require the extendability only into one direction) to an interval J , where the function value is not greater than that in I . The local minimizers are defined similarly. It is important to remark that the above definition differs from the classical definition of local extrema. In the above setting, absolute extrema are also local extrema, moreover, the inner points of an

interval where the function is constant are not necessarily local extrema. For example, the function $x \mapsto \max\{x - 3/4, 0\} + \min\{x - 1/4, 0\}$, $x \in [0, 1]$ has only one local minimizer at $x = 0$ and one local maximizer at $x = 1$, albeit the function is constant zero in $[1/4, 3/4]$.

Theorem 1 [3] *Let us suppose that problem (1) has a unique solution $u \in C^{1,2}(\bar{Q}_T)$. Let us assume that the right-hand side function r satisfies the assumptions*

A1) r is non-decreasing in its fifth variable, that is if $r_5'' > r_5'$ then $r(t, x, r_3, r_4, r_5'') \geq r(t, x, r_3, r_4, r_5')$ for all $(t, x) \in \bar{Q}_T$ and $r_3, r_4 \in \mathbb{R}$.

A2) $r(t, x, r_3, 0, 0) \equiv 0$ for all $(t, x) \in \bar{Q}_T$ and $r_3 \in \mathbb{R}$.

Then the number of the local minimizers (maximizers) of the final time boundary Γ_T is not greater than that of the initial boundary Γ_0 .

3 Decrease of the number of the local extremizers in the explicit Euler numerical solution

First we rewrite function r into a more appropriate form. Let us introduce the one-variable function $g(s) = r(t, x, r_3, sr_4, sr_5)$, where t, x, r_3, r_4, r_5 are fixed constants. Then using Newton–Leibniz rule, the derivatives of compound functions and assumption A2, we have

$$\begin{aligned} r(t, x, r_3, r_4, r_5) &= g(1) = g(0) + \int_0^1 g'(s) ds = \int_0^1 g'(s) ds \\ &= r_4 \underbrace{\int_0^1 \partial_4 r(t, x, r_3, sr_4, sr_5) ds}_{q_4(t, x, r_3, r_4, r_5)} + r_5 \underbrace{\int_0^1 \partial_5 r(t, x, r_3, sr_4, sr_5) ds}_{q_5(t, x, r_3, r_4, r_5)}. \end{aligned}$$

Here $\partial_i r$ denotes the derivative of r with respect to the i th variable. By introducing the notations indicated in the expression, r can be written in the form

$$r(t, x, r_3, r_4, r_5) = r_4 q_4(t, x, r_3, r_4, r_5) + r_5 q_5(t, x, r_3, r_4, r_5). \quad (2)$$

Because of assumption A1, the function q_5 is nonnegative.

The explicit Euler finite difference solution of problem (1) can be constructed as follows. We define the spatial mesh $x_i = i\Delta x$ ($i = 0, 1, \dots, n+1$), where $\Delta x = 1/(n+1)$ and n is a positive integer, and the temporal mesh $t_j = j\Delta t$, $j = 0, 1, \dots, N_{\Delta t}$, where $\Delta t N_{\Delta t} = T$. With these notations, the finite difference solution of problem (1) can be generated by the iteration

$$u_i^{j+1} = \left(\frac{\Delta t (q_5)_i^j}{\Delta x^2} - \frac{\Delta t (q_4)_i^j}{2\Delta x} \right) u_{i-1}^j + \left(1 - 2 \frac{\Delta t (q_5)_i^j}{\Delta x^2} \right) u_i^j + \left(\frac{\Delta t (q_5)_i^j}{\Delta x^2} + \frac{\Delta t (q_4)_i^j}{2\Delta x} \right) u_{i+1}^j, \quad (3)$$

where $i = 1, \dots, n$, $j = 0, 1, \dots, N_{\Delta t} - 1$, u_i^j approximates $u(j\Delta t, i\Delta x)$, the values u_i^0 are computed from the initial condition, $u_0^j = u_{n+1}^j = 0$ and we used the notations $(q_4)_i^j = q_4(t_j, x_i, u_i^j, \Delta u_i^j, \Delta^2 u_i^j)$, $(q_5)_i^j = q_5(t_j, x_i, u_i^j, \Delta u_i^j, \Delta^2 u_i^j)$, where

$$\Delta u_i^j = \frac{u_{i+1}^j - u_{i-1}^j}{2\Delta x}, \quad \Delta^2 u_i^j = \frac{u_{i-1}^j - 2u_i^j + u_{i+1}^j}{\Delta x^2}.$$

The iteration (3) can be written as a vector iteration as $u^{j+1} = X_j u^j$, where $u^j = [u_0^j, \dots, u_{n+1}^j]^T$ and the tridiagonal iteration matrix X_j may depend on the mesh parameters Δt and Δx and even on the vector u^j .

The number of the local extremizers of the iteration vector can be defined as that of the piecewise linear interpolation function of the points $(x_0, u_0^j), \dots, (x_{n+1}, u_{n+1}^j)$. Using Whitney transformations [1, 4], it is shown in paper [2] that if the matrix DXD^{-1} ($D = \text{tridiag}[-1, 1, 0]$) is a totally nonnegative (TN) matrix (meaning that all its minors are nonnegative) then the multiplication by the matrix X cannot increase the number of the local minimizers (maximizers) of a vector. Based on this consideration, we can formulate the discrete analogue of Theorem 1.

Theorem 2 *Let us assume that function r in problem (1) satisfies assumptions A1-A2 (that is it can be written in form (2) with a nonnegative function q_5),*

A3)

$$0 < \Delta x \leq 2 \frac{\min_i (q_5)_i^j}{\max_i |(q_4)_i^j|}$$

(if $\max_i |(q_4)_i^j| = 0$ then there is no upper bound for the mesh size) and

A4)

$$0 < \frac{\Delta t}{\Delta x^2} \leq \frac{1}{4} \min_i \frac{1}{(q_5)_i^j}$$

then the number of the local minimizers (maximizers) of the iteration vector generated by the scheme (3) is not increased in the $u^j \rightarrow u^{j+1}$ step.

Proof: The matrix X_j is tridiagonal. It can be easily shown that the matrix $T_j = DX_j D^{-1}$ is also tridiagonal. It is known [1] that if a tridiagonal matrix is nonnegative and diagonally dominant then it is a TN matrix. Thus, the multiplication with X_j cannot increase the number of the local extrema. The non-negativity of matrix T_j is guaranteed by the assumptions A3-A4. Assumption A3 implies that the offdiagonal elements are nonnegative, while assumption A4 is enough to the dominance of the main diagonal.

Remark 3 *Let us notice that assumptions A3-A4, in the most general case, give an upper bound for the spatial step-size Δx (the spatial mesh should be sufficiently fine) and an upper bound for the mesh-ratio $\Delta t/\Delta x^2$.*

Remark 4 *For the classical linear problem $\partial_t u = \partial_{xx} u$, where $r(t, x, r_3, r_4, r_5) = r_5$, the monotone decrease of the local extrema can be guaranteed by the sufficient condition $\Delta t/\Delta x^2 \leq 1/4$ which is much stricter than the condition of the convergence of the explicit Euler method ($\Delta t/\Delta x^2 \leq 1/2$).*

4 Numerical example

Let us consider problem (1) with $r(t, x, r_3, r_4, r_5) = r_5^3$ (which function trivially satisfies assumptions A1 and A2) and with an initial function that is discretized as shown on the left panel of Figure 1. We have used the spatial mesh size $\Delta x = 1/6$. When we set $\Delta t = 10^{-8}$ and perform 10 iteration steps with method (3), we obtain the approximation

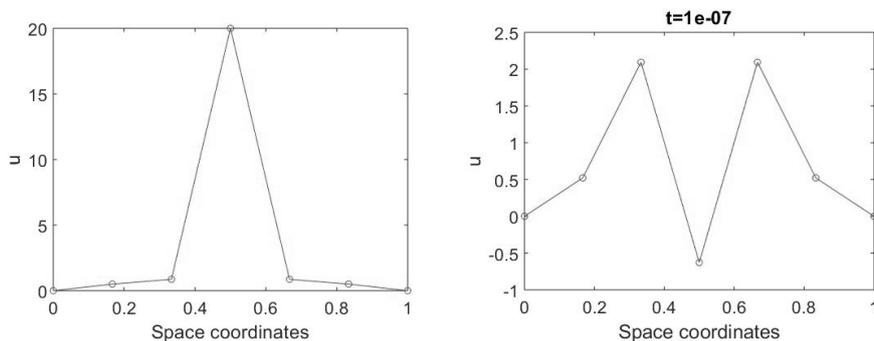


Figure 1: The approximations at the initial time level and at $t = 10^{-7}$ with time step $\Delta t = 10^{-8}$.

at $t = 10^{-7}$ seen on the right panel. We may observe that both the number of the local minimizers and the number of the local maximizers have been increased.

When we use the result of Theorem 2, we have to choose the mesh parameters according to the formula in assumption A4 (q_4 is zero in this case). This gives that if we reach the desired time level $t = 10^{-7}$ with the consecutive time steps $\Delta t = 3.6590 \times 10^{-9}$, $\Delta t = 1.8813 \times 10^{-8}$ and $\Delta t = 8.6939 \times 10^{-8}$ then the approximation will be qualitatively correct. Thus, we have to choose much smaller time step in the first step of the iteration than the time step of the qualitatively wrong iteration and after that much larger time steps are also allowed. This example verifies the result of Theorem 2 and shows its applicability in practice to generate a qualitatively correct solution.

Acknowledgments. This research was supported by the Hungarian Scientific Research Fund OTKA, No. K112157 and SNN125119. The research reported in this paper and carried out at BME has been supported by the NRDI Fund (TKP2020 NC, Grant No. BME-NC) based on the charter of bolster issued by the NRDI Office under the auspices of the Ministry for Innovation and Technology.

References

- [1] **S. M. Fallat, C. R. Johnson**, *Totally Nonnegative Matrices*, Princeton Series in Applied Mathematics, 2011.
- [2] **R. Horváth**, *On some oscillatory properties of finite difference methods for one-dimensional nonlinear parabolic problems*, in preparation.
- [3] **K. Nickel**, *Gestaltaussagen über Lösungen parabolischer Differentialgleichungen*, *Reine Angew. Math.*, **211** (1962), 78–94.
- [4] **A. M. Whitney**, *A reduction theorem for totally positive matrices*, *J. Anal. Math.*, **2** (1952), 88–92.

The method of multiple Richardson extrapolation

Teshome Bayleyegn¹ and Ágnes Havasi²

¹ELTE Eötvös Loránd University, 1117 Budapest, Pázmány Péter s. 1/C, Hungary

²ELTE Eötvös Loránd University, Institute of Mathematics and MTA-ELTE Numerical Analysis and Large Networks Research Group, 1117 Budapest, Pázmány Péter s. 1/C , Hungary

¹sbayleyegn130@gmail.com, ²agnes.havasi@ttk.elte.hu

1 Introduction

Ordinary and partial differential equations are the most important and frequently occurring mathematical models in several areas of applied mathematics. In order to study and understand certain physical, biological, etc. phenomena, such types of equations should be solved, often with a high accuracy and/or within a reasonable computational time. Richardson extrapolation [3] is one of the most powerful numerical techniques which can be used in the efforts to improve the accuracy and performance of underlying numerical methods to solve large and complex problems.

The procedure is based on calculating a suitable linear combination of numerical solutions obtained on two meshes by the same underlying numerical method of order p . This original version of the method is called classical Richardson extrapolation (CRE). The CRE increases the order of accuracy by one if the right-hand side function of the ordinary differential equation to be solved is sufficiently smooth. However, this accuracy is not always sufficient in the applications. The question arises naturally: how can we increase the order even further?

We present a possible generalization of the CRE, which we call multiple Richardson extrapolation (MRE). This method can be combined with any one-step numerical method, e.g., with some Runge–Kutta method, both explicit and implicit. When stiff systems are solved, which frequently arise, e.g., in chemical models, the numerical method should have favourable stability properties on a fixed mesh. In search for an accurate scheme with good absolute stability properties, we will study the absolute stability of the MRE for the simplest Runge–Kutta methods, namely, the first order explicit Euler (EE) and implicit Euler (IE) methods and analyse their stability regions.

2 Classical and multiple Richardson extrapolation

The classical Richardson extrapolation (CRE) method allows us to increase the order p of the underlying method by one. Consider the Cauchy problem for a system of ODE's

$$\begin{cases} y'(t) = f(t, y), & t \in [0, T] \\ y(0) = y_0, \end{cases} \quad (1)$$

where the unknown function y is of the type $\mathbb{R} \rightarrow \mathbb{R}^d$ and $y_0 \in \mathbb{R}^d$. Solve the problem with two different time-step sizes, h and $h/2$, and denote the numerical solutions at time t_n of

the coarse mesh by z_n and w_n respectively. Then the combined solution

$$y_{\text{CRE},n} := \frac{2^p w_n - z_n}{2^p - 1} \quad (2)$$

which is called classical Richardson extrapolation, approximates the exact solution to the order $p + 1$.

The multiple Richardson extrapolation (MRE) is a new procedure [1] obtained by applying CRE to the combined method (CRE + underlying method of order p), and it provides an order of accuracy $p + 2$.

$$y_{\text{MRE},n} := \frac{2^{p+1} y_{\text{CRE}}^{h/2} - y_{\text{CRE}}^h}{2^{p+1} - 1} \quad (3)$$

3 Absolute stability analysis

The absolute stability analysis is based on Dahlquist's scalar test problem

$$\begin{cases} y'(t) = \lambda y(t), & t \in [0, \infty) \\ y(0) = y_0, \end{cases} \quad (4)$$

where $y : \mathbb{R} \rightarrow \mathbb{C}$, $\lambda = \alpha + \beta i \in \mathbb{C}$, $y_0 \in \mathbb{C}$. The exact solution is $y(t) = y_0 \exp(\lambda t)$, $t \in [0, \infty)$, which is bounded iff $\alpha \leq 0$. From a numerical method for stiff problems it is required that the numerical solution of (4) remains bounded for $\alpha \leq 0$ as $t_n \rightarrow \infty$ for any or at least not too small time-steps h . Let $\mu := \lambda h$, then for a one-step method $y_{n+1} = R(\mu)y_n$, where the function R , depending on μ is called stability function of the method. Clearly, the numerical solution remains bounded for the grid points of $[0, \infty)$ iff $|R(\mu)| \leq 1$. The set $S := \{\mu \in \mathbb{C} : |R(\mu)| \leq 1\}$ is called stability region of the method with stability function $R(\mu)$. It is desirable that S is as large as possible, and for stiff systems it should involve \mathbb{C}^- , i.e., the whole left half-plane with the imaginary axis. If $\mathbb{C}^- \subset S$, then the method is called A-stable.

We plotted the stability regions for the EE method as underlying method for different versions of the Richardson extrapolation in Figure 1. The CRE increases the stability region, which becomes even larger for the MRE. The figure also shows the stability region obtained for another possible generalization of CRE, called repeated Richardson extrapolation (RRE, [5]), but the MRE has a larger stability region. For more results for other explicit Runge–Kutta methods see [1]. A larger stability region allows the choice of larger time steps, which improves the efficiency. However, since for the EE method we always get bounded stability regions, the application of MRE is not very helpful when stiff problems are to be solved. Therefore, in the following we investigate the implicit Euler (IE) method as underlying method.

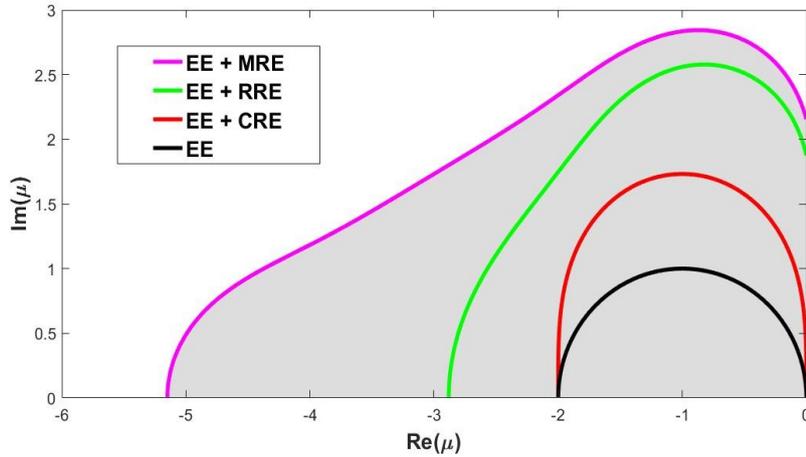


Figure 1: Stability regions for EE as underlying method.

In [2] we analysed the absolute stability function and stability region of the IE + MRE method. The IE and IE + CRE methods have been proved to be A-stable, so their stability regions include the entire left half-plane [4]. We plotted the stability region (in grey) of the IE + MRE method in Fig. 2, which suggests that this method is also A-stable. However, the zoomed picture in Fig. 3 reveals that the combined method IE + MRE is not A-stable. Since the boundary of the stability region extends to the left half-plane, we can have problems with the absolute stability when the matrix (or Jacobian matrix) of the problem to be solved has purely imaginary eigenvalues. It is not recommended to solve such problems with the IE + MRE method. For more details see [2].

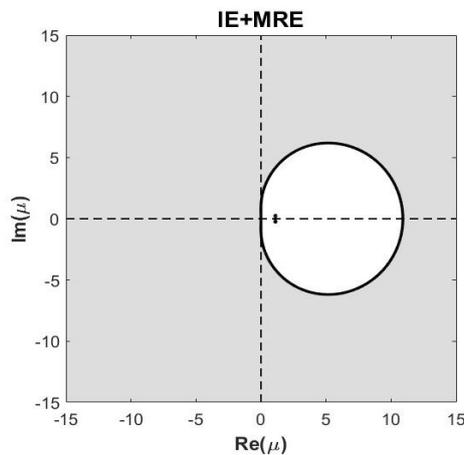


Figure 2: The stability region of IE + MRE.

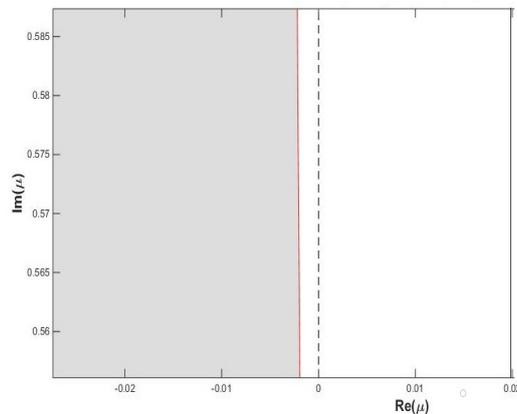


Figure 3: Zoomed detail of the stability region of IE + MRE.

Acknowledgements. “Application Domain Specific Highly Reliable IT Solutions” project has been implemented with the support provided from the National Research, Development and Innovation Fund of Hungary, financed under the Thematic Excellence Programme TKP2020-NKA-06 (National Challenges Subprogramme) funding scheme. This work was completed in the ELTE Institutional Excellence Program (TKP2020-IKA-05) financed by the Hungarian Ministry of Human Capacities. The project has been supported by the European Union, and co-financed by the European Social Fund (EFOP-3.6.3-VEKOP-16-2017-00002), and further, it was supported by the Hungarian Scientific Research Fund OTKA SNN125119.

References

- [1] Bayleyegn, T., Havasi, Á.: Multiple Richardson Extrapolation Applied to Explicit Runge–Kutta Methods, In Ivan Dimov and Stefka Fidanova, editors, *Advances in High Performance Computing*, Springer, Cham, (2021), pp. 262-270.
- [2] Bayleyegn, T., Havasi, Á.: Multiple Richardson Extrapolation and its Combination with the Implicit Euler Method, *Annales Univ. Sci. Budapest., Sect. Math.*, 2020 (Accepted for publication).
- [3] Richardson, L. F.: The Approximate Arithmetical Solution by Finite Differences of Physical Problems Including Differential Equations, with an Application to the Stresses in a masonry dam, *Philosophical Transactions of the Royal Society of London, Series A* 210 (1911), 307-357.
- [4] Zlatev, Z., Dimov, I., Faragó, I., Havasi, Á.: *Richardson Extrapolation - Practical Aspects and Applications*. De Gruyter, 2017.
- [5] Zlatev, Z., Dimov, I., Faragó, I., Georgiev, K., Havasi, Á.: *Stability Properties of Repeated Richardson Extrapolation Applied Together with Some Implicit Runge-Kutta Methods*, Springer, Cham, vol. 11386 (2019).

Operator splitting and Average method

Lívía Boda

Department of Differential Equations, Budapest University of Technology and
Economics, Hungary
bodalive@gmail.com

In mathematics there are a lot of problems which can be described by differential equations of very complicated structure. Most of the time, we cannot produce the exact solution of these complicated problems, so we have to approximate them numerically using some approximating method. In this talk we analyse one of these approximating methods, namely the operator splitting method, which is a widely and successfully used method in numerical analysis. It helps us when we have a very complicated Cauchy problem, which we want to analyse.

We consider the following Cauchy-problem in \mathbb{R}^m

$$\begin{cases} \dot{y}(t) = Ay(t) = \sum_{i=1}^d A_i y(t) & t \in [0, T] \\ y(0) = y_0, \end{cases} \quad (1)$$

where $y : [0, T] \rightarrow \mathbb{R}^m$ is the unknown function, $y_0 \in \mathbb{R}^m$ is the given initial vector, $A_i \in \mathbb{R}^{m \times m}$ ($i = 1, \dots, d$) are matrices.

The exact solution of Cauchy-problem (1) can be written directly as $y(t) = \exp(tA)y(0)$. And our aim is to approximate the exact solution numerically on the grid

$$\omega_h = \{t_n = n \cdot h, h = \frac{T}{N}, n = 0, 1, \dots, N\}.$$

By using operator splitting, we get a series of simpler Cauchy problems which are linked through their initial conditions. By applying this method it can be significantly easier to solve the problem of finding the numerical solution of the original problem.

The two most popular splitting methods include the sequential splitting (SS) and the Strang-Marchuk (SM) splitting.

The algorithm of sequential splitting in case of two subproblems is the following. In this case the decomposition of A is $A = A_1 + A_2$. If we use the sequential splitting to solve (1) in grid ω_h , means the following two Cauchy-problems:

$$\begin{cases} \dot{y}_1(t) = A_1 y_1(t) & t \in [t_i, t_{i+1}] \\ y_1(t_i) = x_{sp}(t_i), \end{cases} \quad \begin{cases} \dot{y}_2(t) = A_2 y_2(t) & t \in [t_i, t_{i+1}] \\ y_2(t_i) = y_1(t_{i+1}). \end{cases}$$

where $i = 0, \dots, n - 1$, and $x_{sp}(t_{i+1}) = y_2(t_{i+1})$.

Remark 1 *The dependence of the functions y_1 and y_2 on i is not indicated.*

The main difference between the sequential and *Strang-Marchuk splitting* is that the latter computes the values in the midpoints of the subintervals.

Remark 2 *The sequential splitting is a first-order method, the Strang-Marchuk splitting is a second-order method.*

For more details we refer to [1].

As an alternative to the classical splitting methods, we introduce the Average Method with sequential splitting (referred to as the AM_{SS} method) which is based on the following idea: dividing the Cauchy problem (1) into d subproblems, using sequential splitting in all possible sequences, calculating the numerical solutions and then taking their arithmetic mean and let it be the numerical solution in ω_h .

Theorem 3 *Solving the Cauchy-problem (1) using sequential splitting for all possible permutations and then averaging the resulting numerical solutions yields a second-order method, i.e.*

$$\exp(h(A_1 + \dots + A_d)) = \frac{\exp(hA_1) \dots \exp(hA_d) + \dots + \exp(hA_d) \dots \exp(hA_1)}{d!} + \mathcal{O}(h^3). \quad (2)$$

If we have d subproblems and use AM_{SS} method to solve Cauchy problem (1), we have to calculate $d!$ numerical solutions. But if we can find a decomposition for Cauchy problem (1) that includes commuting matrices, the number of subproblems can be significantly reduced. Let $A = A_1 + A_2 + \dots + A_d$, and suppose that $\exists i, j \in \mathbb{N}, i \neq j$ such that $[A_i, A_j] = 0$. Then instead of all the $d!$ permutations, we have $d! - (d-1)! = (d-1)(d-1)!$ elements. If the decomposition includes more commuting pairs of matrices, the reduction might be more significant.

We demonstrate that third-order accuracy can be achieved with the AM_{SS} method. Assume that we have the Cauchy-problem (1), with $d = 2$. We then have the following

Theorem 4 *If and only if $A = A_1 + A_2$, and A_1 and A_2 satisfy the condition $\left[A_1, [A_1, A_2] \right] = \left[A_2, [A_1, A_2] \right]$ then*

$$\exp(h(A_1 + A_2)) = \frac{\exp(hA_1) \exp(hA_2) + \exp(hA_2) \exp(hA_1)}{2} + \mathcal{O}(h^4).$$

We investigated the efficacy of the three splitting methods discussed above on a physical problem. The model was chosen because of the structure of the matrices involved, i.e. sparse matrices whose decomposition into a partially commuting set was easy.

A piecewise-linear model of flutter was investigated in [2]. Motivated by this model, we consider the following 4-dimensional Cauchy problem

$$\begin{cases} \dot{\mathbf{x}}(t) &= A_k \mathbf{x}(t), \\ \mathbf{x}(0) &= \mathbf{x}_0. \end{cases} \quad (3)$$

where the affine model equations contain the 3 system matrices ($k = 0, 1, 2$)

$$A_k = \begin{pmatrix} 0 & 1 & 0 & 0 \\ -1 & -(p_1 + p_2\mu c_k) & -\mu^2 c_k p_2 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & c_k \mu & -(p_4 - c_k \mu^2) & -p_3 \end{pmatrix},$$

with the model parameters given in Table 1. and $\mu \in (0, \infty)$ represents the nondimensional wind speed.

Parameter	c_0	c_1	c_2	p_1	p_2	p_3	p_4
Value	5.932	-6.846	2.662	0.1485	0.0147	0.0540	0.2748

Table 1: Parameters of the model

We analyzed a lot of decompositions of matrix A_k , the most important of them is the following which contains commuting matrices:

$$A_k = A_{k(1)} + A_{k(2)} + A_{k(3)},$$

where

$$A_{k(1)} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ -1 & -(p_1 + p_2\mu c_k) & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, A_{k(2)} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -(p_4 - c_k \mu^2) & -p_3 \end{pmatrix},$$

$$A_{k(3)} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & -\mu^2 c_k p_2 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & c_k \mu & 0 & 0 \end{pmatrix}.$$

For the matrices $A_{k(1)}$ and $A_{k(2)}$ $A_{k(1)}A_{k(2)} = \mathbf{0}$ and $A_{k(2)}A_{k(1)} = \mathbf{0}$, so $[A_{k(1)}, A_{k(2)}] = \mathbf{0}$. For more decompositions we refer to [3].

First, we tested the first-order methods, we compared the operation of sequential splitting with the also first-order explicit Euler method. In experiment 1 we used sequential splitting, and in experiment 2 we solve the whole original problem without any splitting, using explicit Euler method. And we got a positive result in terms of runtimes, for the same order of accuracy splitting methods are faster than the full numerical solution. In Table 2. it can be seen that by reducing the step size h , the solvers containing splitting produce the numerical solution 1-2 orders of magnitude faster than the Euler method.

And after that we tested the second-order methods. We compared the Strang-Marchuk and the second-order Average Method with the also second-order improved Euler method. In experiment 3 we use SM splitting, in experiment 4 we use Average Method, solve the subproblems in parallel. And in experiment 5 we solve the whole problem using improved Euler method without any splitting. We got positive result in terms of errors because the errors are two orders of magnitude smaller for the same stepsize h using the splitting method than using the second-order Euler method. It can be seen in Table 3.

h	Experiment 1.	Experiment 2.
1	$7.02 \cdot 10^{-5}$	$2.39 \cdot 10^{-3}$
0.1	$8.44 \cdot 10^{-4}$	$5.32 \cdot 10^{-3}$
0.01	$1.70 \cdot 10^{-3}$	$1.09 \cdot 10^{-2}$
0.001	$1.37 \cdot 10^{-2}$	$7.14 \cdot 10^{-1}$

Table 2: Comparison of runtimes (in seconds) for Experiment 1-3.

h	Experiment 3	Experiment 5
1	$7.20 \cdot 10^{-4}$	$8.21 \cdot 10^{-2}$
0.1	$4.90 \cdot 10^{-7}$	$8.24 \cdot 10^{-5}$
0.01	$4.78 \cdot 10^{-10}$	$7.52 \cdot 10^{-8}$
0.001	$4.76 \cdot 10^{-13}$	$7.43 \cdot 10^{-11}$

Table 3: Comparison of errors in case of experiment 3 and 5

And we got positive result in terms of runtimes too, because the Average Method is about two orders of magnitude faster than the improved Euler method. It can be seen in Table 4. This is a good result for the applicability of the Average Method.

h	Experiment 3	Experiment 4	Experiment 5
1	$4.32 \cdot 10^{-2}$	$1.15 \cdot 10^{-4}$	$8.18 \cdot 10^{-3}$
0.1	$7.55 \cdot 10^{-1}$	$1.01 \cdot 10^{-3}$	$1.96 \cdot 10^{-2}$
0.01	$5.20 \cdot 10^0$	$3.65 \cdot 10^{-3}$	$8.44 \cdot 10^{-2}$
0.001	$1.53 \cdot 10^1$	$1.89 \cdot 10^{-2}$	$1.13 \cdot 10^0$

Table 4: Comparison of runtimes (in seconds) for Experiments 3-5.

By performing several numerical experiments we demonstrated that the benefits of the Average Method are the following:

- easy implementation when d is small,
- provides a second-order approximation solution using a first-order method,
- the numerical solutions of the subproblems can be independently computed, therefore the method can be parallelized.

References

- [1] **I. Farago and A. Havasy** *Operator splittings and their applications*, Nova Science Publ., 2009.
- [2] **T. Kalmar-Nagy, R. Csikja, and T. A. Elgohary** *Nonlinear analysis of a 2-dof piecewise linear aeroelastic system*, Nonlinear Dynamics, vol. 85, no. 2, pp. 739-750, 2016.
- [3] **L. Boda, I. Farago, T. Kalmar-Nagy** *The Average Method is Much Better than Average*, Journal of Computational and Applied Mechanics, Vol. 16, No. 1, pp. 1–20, 2021.

Application of the Carleman linearisation method to partial differential equations

Gabriella Svantnérné Sebestyén

Department of Differential Equations, Budapest University of Technology and Economics
gabriella.sebestyen@gmail.com

Abstract

In this article we have studied the application of the Carleman linearisation method to partial differential equations. We have analysed it on the Poisson's equation, which has important role at physics and engineering. We have investigated the classical finite difference method and a new method, namely the method of lines. In this case we replace the partial differential equation by a second order system of ordinary differential equations with boundary values. We have investigated some numerical techniques for this problem including the Carleman linearisation method.

1 Introduction

The Poisson's equation is a second order partial differential equation, it describes an physical phenomena in electrostatics [1], [2]. It is the generalization of the Laplace's equation, which is also an important equation in physics. In this article we investigate the following two-dimensional Poisson's equation

$$-\left(\frac{\partial^2 u}{\partial x^2}(x, y) + \frac{\partial^2 u}{\partial y^2}(x, y)\right) = f(x, y), \quad (x, y) \in (0, L) \times (0, L) \quad (1)$$

where $u(x, y) : \mathbb{R}^2 \rightarrow \mathbb{R}$ is the unknown function and $f(x, y) : \mathbb{R}^2 \rightarrow \mathbb{R}$ is a given function. We suppose that $u(x, y)$ satisfies the homogeneous Dirichlet boundary condition:

$$u|_{\partial\Omega} = 0. \quad (2)$$

The exact solution of the Poisson's equation can not be determined in general, so we use numerical methods to solve the equation (1)–(2). In the following we summarize the numerical methods for this problem.

2 Finite difference method

The finite difference method is a well-known numerical method to solve the Poisson's equation [3], [5]. We replace the derivatives in the equation (1) by finite differences. We define sequences of meshes in the following way:

$$\begin{aligned} x_i &= ih, & i &= 0, 1, \dots, M+1, & h &= \frac{L}{M+1}, \\ y_j &= jh, & j &= 0, 1, \dots, M+1, & h &= \frac{L}{M+1}. \end{aligned} \quad (3)$$

At an inner point (x_i, y_j) , we approximate the second derivatives by the second order central difference as follows

$$\begin{aligned}\frac{\partial^2 u}{\partial x^2}(x, y) &\approx \frac{u(x_{i-1}, y_j) - 2u(x_i, y_j) + u(x_{i+1}, y_j)}{h^2}, \\ \frac{\partial^2 u}{\partial y^2}(x, y) &\approx \frac{u(x_i, y_{j-1}) - 2u(x_i, y_j) + u(x_i, y_{j+1})}{h^2}.\end{aligned}$$

We denote $U_{i,j}$ the approximation of the solution at the point (x_i, y_j) , then the approximation of the Poisson's equation (1) has the form

$$-\left(\frac{U_{i-1,j} - 2U_{i,j} + U_{i,j+1}}{h^2} + \frac{U_{i,j-1} - 2U_{i,j} + U_{i,j+1}}{h^2}\right) = f(x_i, y_j) \quad (4)$$

for $i, j = 1, 2, \dots, M$. At an inner point (x_i, y_j) the difference scheme depends on the following five points (x_{i-1}, y_j) , (x_{i+1}, y_j) , (x_i, y_j) , (x_i, y_{j+1}) , (x_i, y_{j-1}) . The approximation (4) of the Poisson's equation can be written into a system of linear equations.

3 Carleman linearisation method

In the following we apply the Carleman linearisation method to solve the problem (1)–(2). The Carleman linearisation method has been developed to transform sets of polynomial ordinary differential equations into an infinite dimensional linear system [4]. At first step we replace the partial differential equation by an ordinary differential equation. Let

$$u(x, y_i) \approx Y_i(x) \quad (5)$$

be the approximation of the solution of the Poisson's equation. Then the partial differential equation (1)–(2) can be written into the following boundary value problem

$$-\left(Y_i''(x) + \frac{Y_{i+1}(x) - 2Y_i(x) + Y_{i-1}(x)}{h^2}\right) = f_i(x), \quad x \in (0, L), \quad i = 1, 2, \dots, M. \quad (6)$$

We introduce new functions in the following way

$$\begin{aligned}v_1(x) = Y_1(x) \quad v_2(x) &= Y_1'(x) \\ v_3(x) = Y_2(x) \quad v_4(x) &= Y_2'(x) \\ &\vdots \\ v_{2M-1}(x) = Y_M \quad v_{2M}(x) &= Y_M'(x)\end{aligned} \quad (7)$$

and the problem (6) can be transformed into a system of differential equations

$$\begin{aligned}
 v_1' &= v_2 \\
 -\left(v_2' + \frac{v_3 - 2v_1}{h^2}\right) &= f_1(x) \\
 v_3' &= v_4 \\
 -\left(v_4' + \frac{v_5 - 2v_3 + v_1}{h^2}\right) &= f_2(x) \\
 &\vdots \\
 v_{2M-1}' &= v_{2M} \\
 -\left(v_{2M}' + \frac{-2v_{2M-1} + v_{2M-3}}{h^2}\right) &= f_M(x)
 \end{aligned} \tag{8}$$

with the following initial conditions

$$\begin{aligned}
 v_1(0) = v_3(0) = \dots = v_{2M-1}(0) &= 0 \\
 v_2(0) = c_1, v_4(0) = c_2, \dots, v_{2M}(0) &= c_M,
 \end{aligned} \tag{9}$$

where c_1, c_2, \dots, c_M are unknown values. Our aim is to determine the $c_i, i = 1, \dots, M$ values that satisfy the following boundary conditions

$$v_1(L) = v_3(L) = \dots = v_{2M-1}(L) = 0. \tag{10}$$

In the following we investigate the numerical methods for the system of ordinary differential equations (8)–(10).

- The problem (6) is an special second order problem, because the differential equation is linear. We solve the problem (6) with two different initial conditions. We denote the solutions

$$\begin{aligned}
 Y_{i,1} &= Y_i(x, C_1) \\
 Y_{i,2} &= Y_i(x, C_2),
 \end{aligned}$$

$i = 1, 2, \dots, M$ when C_1 and C_2 are different initial values. Let

$$w_i(x) = \lambda_i Y_{i,1}(x) + (1 - \lambda_i) Y_{i,2}(x) \tag{11}$$

be the linear combination of the two solutions. It can be shown that $w_i(x)$ satisfies the problem (6). If we substitute the boundary conditions we get the following relationships at the point zero and L : $w_i(0) = 0$ and $w_i(L) = \lambda w_{i,1}(L) + (1 - \lambda) w_{i,2}(L)$. We choose the parameter λ , that the boundary condition is satisfied: $w_i(L) = 0$. We can determine λ_i in the following way

$$\lambda_i = \frac{-Y_{i,2}(L)}{Y_{i,1}(L) - Y_{i,2}(L)}. \tag{12}$$

If we determine λ_i , then the solution of the problem (6) can be determined as well.

- The system of differential equations (8) can be written in matrix form

$$\begin{pmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \\ \vdots \\ v_{2M} \end{pmatrix}' = \begin{pmatrix} 0 & 1 & 0 & 0 & \dots & 0 \\ \frac{2}{h^2} & 0 & -\frac{1}{h^2} & 0 & \dots & 0 \\ 0 & 0 & 0 & 1 & 0 & \dots \\ -\frac{1}{h^2} & 0 & \frac{2}{h^2} & 0 & -\frac{1}{h^2} & \dots \\ & & & \vdots & & \\ \dots & 0 & -\frac{1}{h^2} & 0 & \frac{2}{h^2} & 0 \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \\ \vdots \\ v_{2M} \end{pmatrix} - \begin{pmatrix} 0 \\ f_1(x) \\ 0 \\ f_2(x) \\ \vdots \\ f_M(x) \end{pmatrix}. \quad (13)$$

and the initial vector is the following

$$(0, c_1, 0, c_2, 0, \dots, 0, c_M)^T. \quad (14)$$

We can apply the Carleman linearisation method to the system (13). The Carleman linearisation method transforms this system of differential equations into an infinite dimensional linear system. We introduce the vector V , which consists of the polynomial of v_i in the following way

$$V = (v_1, \dots, v_{2M}, v_1^2, \dots, v_{2M}^2, v_1 v_2, v_1, v_1 v_3, \dots)^T. \quad (15)$$

By omitting the higher order terms, we get the following system of differential equations

$$\frac{dV}{dt} = C_N V + F \quad (16)$$

where C_N is the Carleman matrix and $F = (0, f_1, 0, f_2, \dots, 0, f_M, \dots)^T$. We can use different techniques to determine the approximation of the solution, for example operator splitting method or exponential integrator method.

Future plans

In the future we plan to apply the previous techniques to determine the numerical solution of partial differential equations. It will be interesting to compare the results with the finite difference method. We would like to apply the Carleman linearisation method to non linear partial differential equations and other kind of boundary conditions too.

References

- [1] **Jackson J. D.**, *Classical Electrodynamics Third Edition*, 3rd Edition, Wiley, 1998.
- [2] **Kraus J. D., Fleisch D. A.**, *Electromagnetics with Applications, 5th Edition*, McGraw–Hill, 1999.
- [3] **Lapidus L., Pinder G. F.**, *Numerical Solution of Partial Differential Equations in Science and Engineering*, Wiley, 1982.
- [4] **Steeb, W.H., Wilhelm, F.**, *Non-linear autonomous system of differential equations and Carleman linearization procedure J. Math Anal. Appl*, **44** (1980), 601–611.
- [5] **Strikwerda J.**, *Numerical Solution of Partial Differential Equations in Science and Engineering*, SIAM, second edition, 2007.

Numerical methods for space-dependent epidemic models

**Bálint Takács^{1,†}, Róbert Horváth^{2,‡}, István Faragó^{1,2,*},
Yiannis Hadjimichael³**

¹Eötvös Loránd University, Budapest, Hungary

²Budapest University of Technology and Economics, Hungary

³Weierstrass Institute, Berlin, Germany

[†]`takacs.balint.mate@gmail.com`, [‡]`rhorvath@math.bme.hu`,

^{*}`faragois@math.bme.hu`

The SIR model, first introduced by Kermack and McKendrick [7] can be used to describe any process in which some property is passed among a group of individuals. During the process, we distinguish three classes: the first one (labeled by S) contains the ones which has not acquired the property yet, the second (denoted by I) has those which have the property and have the ability to pass it on to others, and the last one (class R) contains the ones which had the property, but they cannot transmit it any more. Such processes include epidemics (the process the model first was introduced for) or other biological phenomena like a fire in a forest. The model mentioned above can be written in the following form, in which a, b and c describe the infection rate, the rate of recovery and the effect of vaccination, respectively.

$$\begin{cases} \frac{dS(t)}{dt} = -aS(t)I(t) - cS(t), \\ \frac{dI(t)}{dt} = aS(t)I(t) - bI(t), \\ \frac{dR(t)}{dt} = bI(t) + cS(t). \end{cases} \quad (1)$$

The original system of ordinary differential equations can be extended introducing a spatial dependence, resulting in a system of partial integro-differential equations. Let us suppose that an individual can only be affected by an ill person if they are close to each other, and this effect also weakens as they get further. Because of this, we can introduce a non-negative function

$$G(x, y, r, \theta) = \begin{cases} g_1(r)g_2(\theta), & \text{if } (\bar{x}(r, \theta), \bar{y}(r, \theta)) \in B_\delta(x, y), \\ 0, & \text{otherwise,} \end{cases} \quad (2)$$

that describes the effect of a single point (x, y) in a δ -radius neighborhood $B_\delta(x, y)$, and set $\bar{x}(r, \theta) = x + r \cos(\theta)$ and $\bar{y}(r, \theta) = y + r \sin(\theta)$. The function $G(x, y, r, \theta)$ demonstrates how healthy individuals at points $(\bar{x}(r, \theta), \bar{y}(r, \theta))$ are infected by the center point (x, y) , where $r \in [0, \delta]$ is the distance from the center and $\theta \in [0, 2\pi)$ is the angle. Here we assume that the right-hand-side of (2) is separable. The effect of the point (x, y) depending on the distance from the center is described by $g_1(r)$: a decreasing, non-negative function that is zero for values $r \geq \delta$ (since there is no effect outside $B_\delta(x, y)$). The bounded non-negative

function $g_2(\theta)$ characterizes the part of the effect depending on the angle, i.e., the direction in which the center is compared to point $(\bar{x}(r, \theta), \bar{y}(r, \theta))$. Let us use the notation

$$G_I(t, x, y) = \int_0^\delta \int_0^{2\pi} g_1(r) g_2(\theta) I(t, \bar{x}, \bar{y}) r d\theta dr.$$

Then, our equation can be written in the form [6]

$$\begin{cases} \frac{\partial S(t, x, y)}{\partial t} = -S(t, x, y)G_I(t, x, y) - cS(t, x, y), \\ \frac{\partial I(t, x, y)}{\partial t} = S(t, x, y)G_I(t, x, y) - bI(t, x, y), \\ \frac{\partial R(t, x, y)}{\partial t} = bI(t, x, y) + cS(t, x, y). \end{cases} \quad (3)$$

In the talk we discussed several possible further extensions of this model, like the inclusion of a constant delay α or the addition of diffusion of the individuals, see the following systems.

$$\begin{cases} \frac{\partial S(t, x, y)}{\partial t} = -S(t, x, y)G_I(t - \alpha, x, y) - cS(t, x, y), \\ \frac{\partial I(t, x, y)}{\partial t} = S(t, x, y)G_I(t - \alpha, x, y) - bI(t, x, y), \\ \frac{\partial R(t, x, y)}{\partial t} = bI(t, x, y) + cS(t, x, y). \end{cases} \quad (4)$$

$$\begin{cases} \frac{\partial S(t, x, y)}{\partial t} = -S(t, x, y)G_I(t, x, y) - cS(t, x, y) + D_S \Delta S(t, x, y), \\ \frac{\partial I(t, x, y)}{\partial t} = S(t, x, y)G_I(t, x, y) - bI(t, x, y) + D_I \Delta I(t, x, y), \\ \frac{\partial R(t, x, y)}{\partial t} = bI(t, x, y) + cS(t, x, y) + D_R \Delta R(t, x, y). \end{cases} \quad (5)$$

Here D_S , D_I and D_R denote the diffusion parameters corresponding to the different species.

In the talk we showed several numerical models arising from these continuous systems. We used different techniques to discretize our problem in space: first, we approximate the integral term in the equations using either an Elhay-Kautsky [2, 5, 9] or a Gauss-Legendre quadrature \mathcal{Q} involving the transformation of the circle onto a square [8] with positive coefficients $w_{i,j}$. It turned out that for arbitrary nonlinear functions the latter works better [10].

Then, we introduce a spatial mesh on our rectangular domain with stepsize h . In the case of system (5) we also need to approximate the Laplace operator: this can be done by using a central difference scheme

$$\begin{aligned} \Delta S(t, x_k, y_l) &\approx \mathcal{D}_0^2(S_{k,l}(t)) = \\ &= \frac{S_{k+1,l}(t) + S_{k-1,l}(t) - 4S_{k,l}(t) + S_{k,l+1}(t) + S_{k,l-1}(t)}{h^2}. \end{aligned}$$

After all these, we get the following system of ordinary differential equations for system (5) (the case (3) is when $D_S = D_I = D_R = 0$):

$$\begin{cases} \frac{dS_{k,l}(t)}{dt} = -S_{k,l}(t)T_{k,l}(t, \mathcal{Q}(x_k, y_l)) - cS_{k,l}(t) + D_S \mathcal{D}_0^2(S_{k,l}(t)), \\ \frac{dI_{k,l}(t)}{dt} = S_{k,l}(t)T_{k,l}(t, \mathcal{Q}(x_k, y_l)) - bI_{k,l}(t) + D_I \mathcal{D}_0^2(I_{k,l}(t)), \\ \frac{dR_{k,l}(t)}{dt} = bI_{k,l}(t) + cS_{k,l}(t) + D_R \mathcal{D}_0^2(R_{k,l}(t)), \end{cases}$$

and in the case of (4):

$$\begin{cases} \frac{dS_{k,l}(t)}{dt} = -S_{k,l}(t)T_{k,l}(t - \alpha, \mathcal{Q}(x_k, y_l)) - cS_{k,l}(t), \\ \frac{dI_{k,l}(t)}{dt} = S_{k,l}(t)T_{k,l}(t - \alpha, \mathcal{Q}(x_k, y_l)) - bI_{k,l}(t), \\ \frac{dR_{k,l}(t)}{dt} = bI_{k,l}(t) + cS_{k,l}(t), \end{cases}$$

where

$$T_{k,l}(t, \mathcal{Q}(x_k, y_l)) := \sum_{(x_{i,j}, y_{i,j}) \in \mathcal{Q}(x_k, y_l)} w_{i,j} g_1(r_i) g_2(\theta_j) \tilde{I}(t, x_{i,j}, y_{i,j}).$$

The reason for the tilde notation above the function I is that it might happen that the point $(x_{i,j}, y_{i,j}) \in \mathcal{Q}(x_k, y_l)$ is not part of the spatial mesh: because of this, we use some (positivity preserving) interpolation (e.g. bilinear or pchip [1, 3]).

Then we solve the above system of ordinary (or delayed) differential equations using a Runge-Kutta method. The main aim of these numerical methods is not only to approximate the analytic solution in a sufficiently large order, but also to have such a numerical scheme which (by using a sufficiently chosen step size) preserves the properties of the original continuous system: the positivity of solutions, the conservation of the total mass of the species and also the monotonicity properties of S and R in the cases of (3) and (4).

Theorem 1 *Consider an explicit Runge-Kutta method with SSP coefficient $\mathcal{C} > 0$ [4] and applied to our time-dependent problem with non-negative initial data. Then the qualitative properties mentioned above hold for the numerical solution if the corresponding condition holds:*

- For system (3) [10]

$$\tau \leq \mathcal{C} \min \left\{ \min \frac{1}{\widehat{T} + c}, \frac{1}{b} \right\}.$$

- For system (4) [11]

$$\tau := \frac{\alpha}{m} \leq \mathcal{C} \min \left\{ \min \frac{1}{\widehat{T} + c}, \frac{1}{b} \right\}.$$

- For system (5):

$$\tau \leq \mathcal{C} \min \left\{ \min \frac{1}{\widetilde{T} + c + \frac{4}{h^2} D_S}, \frac{1}{b + \frac{4}{h^2} D_I} \right\}.$$

Here $\widehat{T} := \mathcal{H}(M)$ and $\widetilde{T} := \mathcal{H}(\widetilde{M})$, where

$$M = \max_{(x_k, y_l) \in \mathcal{G}} \{S(0, x_k, y_l) + I(0, x_k, y_l) + R(0, x_k, y_l)\},$$

$$\widetilde{M} = \sum_{(x_k, y_l) \in \mathcal{G}} S(0, x_k, y_l) + I(0, x_k, y_l),$$

and \mathcal{H} is the operator coming from the spatial discretization of the integral term in a way that $T^n = \mathcal{H}(I^n)$ (where matrices T^n and I^n contain the approximations of $T(t_n, \mathcal{Q}(\cdot, \cdot))$ and $I(t_n, \cdot, \cdot)$). Also, the SSP coefficient is 1 for the 2nd and 3rd order methods, and is 6 for the 4th order one.

Therefore, we could show that if we use a sufficiently small timestep, then the numerical scheme preserves the required properties.

References

- [1] DOUGHERTY, R. L., EDELMAN, A. S., HYMAN, J. M., *Nonnegativity-, monotonicity-, or convexity-preserving cubic and quintic Hermite interpolation*, Math. Comp., 52 (1989), pp. 471–494, <https://doi.org/10.2307/2008477>, <https://doi.org/10.2307/2008477>.
- [2] ELHAY, S., KAUTSKY, J., *Algorithm 655: Iqpack: Fortran subroutines for the weights of interpolatory quadratures*, ACM Trans. Math. Software, 13 (1987), pp. 399–415.
- [3] FRITSCH, F. N., CARLSON, R. E., *Monotone piecewise cubic interpolation*, SIAM J. Numer. Anal., 17 (1980), pp. 238–246, <https://doi.org/10.1137/0717021>, <https://doi.org/10.1137/0717021>.
- [4] GOTTLIEB, S., KETCHESON, D. I., SHU, C-W., *Strong stability preserving Runge-Kutta and multistep time discretizations*, World Scientific Publishing Co. Pte. Ltd., Hackensack, NJ, 2011, <https://doi.org/10.1142/7498>, <http://dx.doi.org/10.1142/7498>.
- [5] KAUTSKY, J., ELHAY, S., *Calculation of the weights of interpolatory quadratures*, Numer. Math., 40 (1982), pp. 407–422, <https://doi.org/10.1007/BF01396453>, <https://doi.org/10.1007/BF01396453>.
- [6] KENDALL, D.G., *Mathematical models of the spread of infection*, In : Mathematics and Computer Science in Biology and Medicine. H.M.S.O., London, 1965, pp. 213 – 225.
- [7] KERMAK, W.O., MCKENDRICK, A.G., *A contribution to the mathematical theory of epidemics*, Proc. R. Soc. A: Math. Phys. Eng. Sci. 115 (772) (1927) pp. 235–240.
- [8] MA, J., ROKHLIN, V., WANDZURA, S., *Generalized Gaussian quadrature rules for systems of arbitrary functions*, SIAM Numer. Anal. Vol. 33, No. 3 (1996), pp. 971–996, .
- [9] MARTIN, R. S., WILKINSON, J. H., *Handbook Series Linear Algebra: The implicit QL algorithm*, Numer. Math., 12 (1968), pp. 377–383, <https://doi.org/10.1007/BF02161360>, <https://doi.org/10.1007/BF02161360>.
- [10] TAKÁCS, B., HADJIMICHAEL, Y., *High Order discretization methods for spatial dependent SIR models*, <https://arxiv.org/abs/1909.01330>
- [11] TAKÁCS, B., FARAGÓ, I., HORVÁTH, R., REPOVŠ, D., *Qualitative properties of space-dependent SIR models with constant delay and their numerical solutions*, Numerical Methods for Partial Differential Equations, submitted.

Section:

The ubiquitous machine learning – bridging science and business

Organizer: András Lukács

Invited talk:

Szabolcs Biró and Szilárd Varró: Machine Learning Use Cases in Manufacturing

Contributions:

- Bálint Csanády and András Lukács: 1D Convolutional Neural Networks for Diacritics Restoration
- Gábor Hidy and András Lukács: Nucleus classification with neural networks
- Gellért Károlyi and András Lukács: Transfer learning for medical image classification
- Melinda Kiss, Adrián Csiszárík, Ákos Matszangosz, Balázs Maga and Dániel Varga: Global Sinkhorn Autoencoder - Optimal transport on the latent representation of the full dataset
- Péter Marton, Norbert Bicskei and András Lukács: Machine Learning Algorithms for MOD Lapse at renewal

Machine Learning Use Cases in Manufacturing

Szabolcs Biró, Szilárd Varró

Hiflylabs Zrt., Infominero Kft.

birsza@gmail.com, varro.szilard@gmail.com

Introduction

Industrial digitalization and continuously growing data assets raised the demand for machine learning methodologies in many different use cases. In order to demonstrate the complexity of ML based solutions in manufacturing, three such use cases will be covered briefly.

Alloy Wheel Pitting - Root Cause Analysis

In alloy wheel production, investigating manufacturing failures – like in all manufacturing processes – is highly important task in order to reduce rework and waste ratios. The factory in question has a complex process of producing wheels in three main phases (forging, machining and finishing) with more than 20 consecutive steps (e.g. press stations, polishing, ball burnishing etc.). In addition, there are different lines in production and more different wheel types and sizes. The failure (called pitting) looks like a micro asteroid on the surface of the wheel, and this problem had been present for a decade without knowing the reason exactly. The challenge can be considered as a binary classification problem (since the failure – not failure dichotomy), but after aggregating sensory and other data for a daily time window this dichotomy transforms to a continuous probability measure in case of the target variable. The daily aggregation of per-second data was necessary because the identification per wheel is not available for the whole process thus, we can investigate averaged daily operation measurements and machine setups and their correlation to the average pitting rate. Analytical base tables had been constructed for the different wheel types and production lines (in a daily aggregation level). The following topics were covered by variables: materials (e.g. source of material), methods (e.g. change in polishing recipe), measurements (e.g. polishing pressure), machines (e.g. maintenance logs), environment (e.g. outside temperature), personnel (e.g. operator). Analytical base tables helped engineers and data experts to talk through univariate correlations in order to explore possible root causes and to focus on specific areas and come up with new ideas for explanatory variables. Variable selection based on calculated information value. An example is demonstrated on the left side of Figure 1, we can read that bigger polishing pressure is riskier. After finalizing analytical base tables simple transparent machine learning algorithm (3-layer decision tree) was applied to capture most important effects of pitting failure and to collect threshold values for the right and wrong ranges on machine setups and other factors as well. AUC values for the models are listed on the right side of Figure 1.

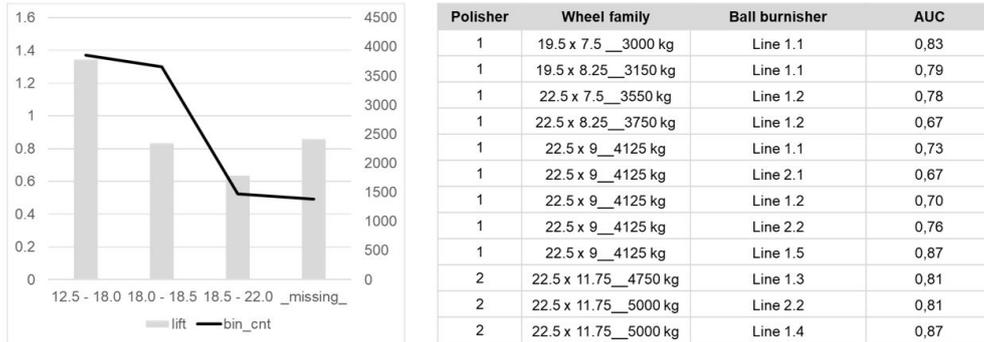


Figure 1: Demonstration of univariate insights (left) and model precisions (right)

After modelling a test production is defined in order to validate findings, the setup for the machines changed based on the decision tree rules (modification of settings and recipes towards the riskier range – but within predefined boundaries). The goal was to artificially create wheels with pitting. The test production had to be canceled after 100 wheels because of the heavy pitting.

Pump failure prediction – Refinery predictive maintenance

In a stage of a refinery subprocess two cooling oil pumps help to cool down reactors, one pump from the two must work, otherwise the system must be shut down. These pumps had leakages on the sealings more frequently than expected. Several years of sensor tag and other data were available: continuous measurement data regarding to the pumps, measurement data about the reactors and other key stations of the process, maintenance logs, quality measurements of input material. The main task was to develop a machine learning model that can be used to predict next failure 20 days in advance. At the beginning we focused on sensor tags directly from pumps (vibration, temperature, etc.), but because of the alternating usage of the two pumps this data could not be used, namely not enough data were produced prior to failures. Besides this, extensive exploratory analysis suggested temporal changes on system level over time thus system level sensor tags had to be analyzed deeper. There was no failure for a year, therefore, a model which separates this period from the years with failures could reveal process parameters which were responsible for failures. Random Forest classifier was fitted to find the tags which separate this year from other periods. After finding important features K-Means clustering algorithm was used to separate days with sensor tags proven to be important in the previous Random Forest model. This way 3 periods could be defined with different system level environment and different failure frequencies. Finally, a random forest regressor was fitted on system level sensor tags in order to predict operating time between failures (this was our target variable), thus operating time until failure could be calculated (per pump). Explanatory variables was calculated on an hourly level with 24 hour sliding window (for all system level sensor tags mean, minimum and maximum were calculated). Next figure shows the prediction of the model ins a simulation, the model is recalibrated after every failure. Failure alert is triggered when predicted operating time until failure is below 20

days. Blue line shows the operating time between failures, which reflects to the status of the system, red line shows operating time since last failure (calculated based on factual flow on pumps). The prediction is a subtraction of these previous two quantities. Green line shows the factual values, it demonstrates that the model had a high precision after simulation (accuracy above 90%).

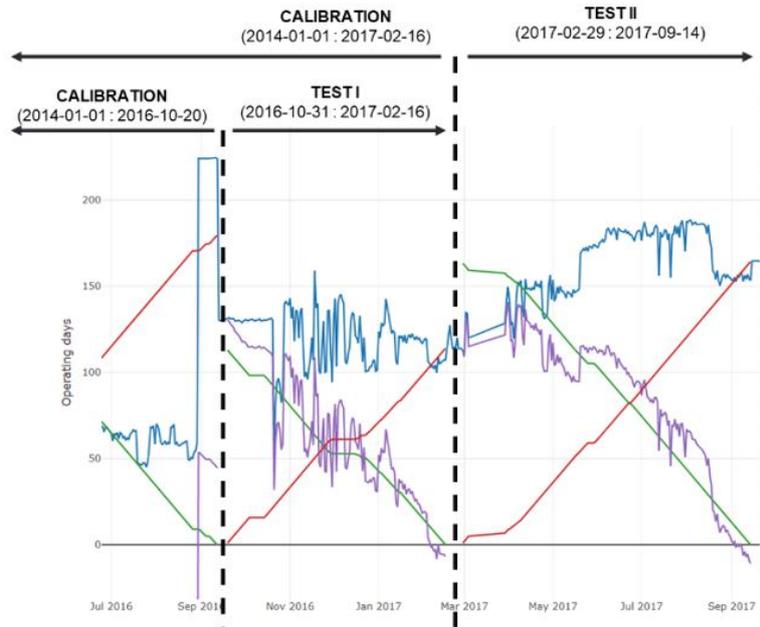


Figure 2: Predictions of pump failure

AI driven 3D aerodynamics correction model

In aerodynamics the calculation of the force distribution on a three-dimensional surface is computationally very intensive, it takes days of calculation for a given geometrical and environmental setup. In practice the calculations are made by using simplified model which neglects several important effects of the real flow experienced in nature, however for many engineering applications this simplified representation has all of the properties that are important from the engineering point of view. The model simplification causes inaccurate results, which leads to inefficiencies in the real application e.g. greater resistance, less lift force, worse rotor efficiency, worse maneuverability, worse flying safety. In our R&D project we moved one step further from a simplified two-dimensional wing model to a machine learning model which is able to make corrected predictions to achieve more accurate three-dimensional results. The process flow from CAD exchange format to prediction is fully automatized, lacks complex three-dimensional analytical calculations, runs in a few seconds and the accuracy is like the slower analytical solutions. The basic geometrical properties are extracted from STEP CAD exchange format. To describe complex geometries with a few parameters, we used Bézier and B-spline curves. There are different neutral files available in CAD software. Some of the neutral files are IGES, STEP, DXF,

STL files etc. STEP and IGES are most popular. STEP is intended for product data exchange, whereas IGES is for geometry data exchange. The description of product data for mechanical parts has been standardized by ISO10303 and different protocols are available in STEP. The STEP file is given as input to the developed program. The developed feature recognition program starts searching the STEP file with a string CLOSED_SHELL and it ends at a string CARTESIAN_POINT. In between various strings such as ADVANCED_FACE, FACE_OUTER_BOUND etc. are searched in a hierarchical manner. The Vortex lattice method, (VLM), is a numerical method used in computational fluid dynamics. VLM models an aircraft surface into infinite number of vortices to calculate lift curve slope, induced drag and force distribution. The VLM models the lifting surfaces, such as a wing, of an aircraft as an infinitely thin sheet of discrete vortices to compute lift and induced drag. We used 2D VLM results, geometrical parameters and operational conditions as model input and 3D VLM analytical results as model output/target. Our model used for predictions was a simple feedforward neural network with 2 fully connected hidden layers. The difficulty of the project was not the modeling, rather the data extraction, preparation and transformation. We were able to reproduce the results of the analytical calculations with acceptable accuracy by a relatively simple AI model. The analytical calculations ran several weeks for a few thousand use cases, the AI model was able to make the predictions in less than a minute after training (the training took a few minutes as well).

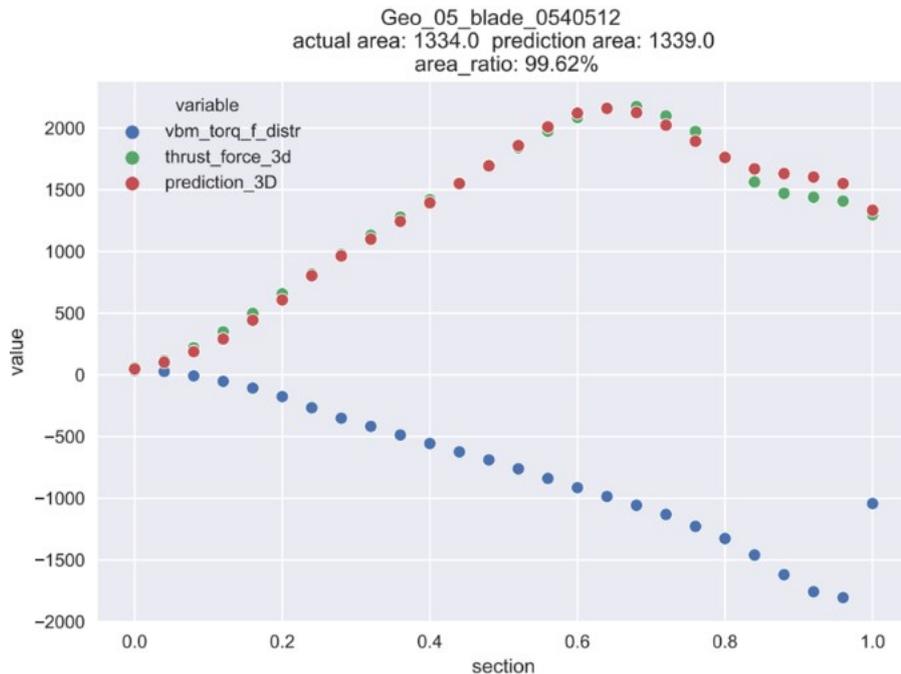


Figure 3: Model accuracy (X-axis demonstrates the position of the wing, Y-axis shows the force while green dots shows factual values, red dots shows prediction)

1D Convolutional Neural Networks for Diacritics Restoration

Bálint Csanády András Lukács

Eötvös Loránd University, Institute of Mathematics
Department of Computer Science, AI Research Group
{csbalint, lukacs}@cs.elte.hu

Abstract

Diacritics restoration became a ubiquitous task in the Latin-alphabet-based English-dominated Internet language environment. In this article, we describe a small footprint 1D convolution-based solution, even running in a web browser, which surpassed the performance of similarly sized models in the case of the Hungarian language.

1 Introduction

In many languages characters are often derived from a base alphabet using *diacritical marks*. The goal of *diacritics restoration* is to restore diacritical marks given an input text without the proper marks. Diacritics restoration is a practical task on the internet, where the fact that computers were initially built with the base Latin alphabet in mind, still shows.

This task is typically modeled as a sequence labeling problem. We present a language-independent method for automatic diacritic restoration using a neural architecture based on 1D convolutions, the so called Acausal Temporal Convolutional Neural Networks (A-TCNs). Other approaches include BiLSTM based solutions [7]. Models based on A-TCN have a comparable performance to BiLSTM-s [2].

We mainly focus on the Hungarian language, where the characters which can receive a diacritic marks are exactly the vowels (e.g. $u \mapsto \{u, \acute{u}, \ddot{u}, \check{u}\}$). For Hungarian the current state of the art is reported by Laki et al. [6] and is achieved by neural machine translation. Our main contribution is a trained model, which runs locally in the browser, allowing client-side inference. We compared our model with Hunaccent [1], since both models have a size of around 10MB. Our approach outperformed Hunaccent by a large margin.

2 Approach

In our research the diacritics restoration problem was modeled as a sequence to sequence task on characters. To solve this seq2seq problem we considered Temporal Convolutional Neural Networks (TCNs). TCNs are a generic family of models, notable examples include WaveNet [12]. Our specific choice is a 1D fully convolutional network from [3], where the convolutions are causal, they convolve output at time t with elements from time $t - 1$ and earlier. To increase the effective size of the convolutional window, the network is built with convolutions with dilation factors [13] exponentially increasing by the depth of the network (Figure 1).

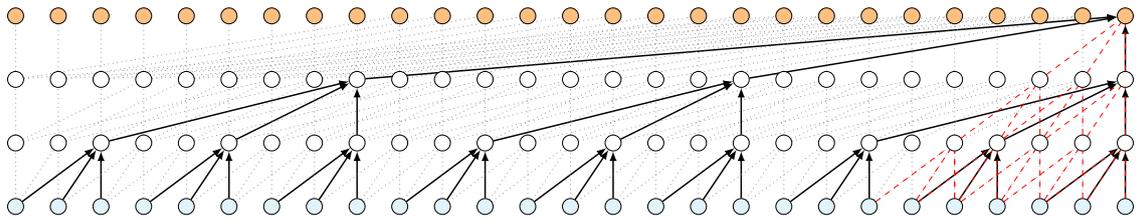


Figure 1: TCN architecture (kernel size: 3, dilation factors: 1,3,9). Red dashed: without dilation.

TCNs also have residual connections [5]. A residual block contains a series of transformations, the result of which are then added to the input. The transformation consists of a dilated convolution, followed by a normalization layer, activation function, and dropout [11]. This is repeated b times (typically $b = 2$).

TCNs work well for applications where information flow from the future is not permitted. For diacritics restoration it is essential to incorporate future context as well as past context. In order to do this, we have to slightly modify the base TCN architecture as seen in Figure 2. These acausally modified TCNs are called A-TCNs [2].

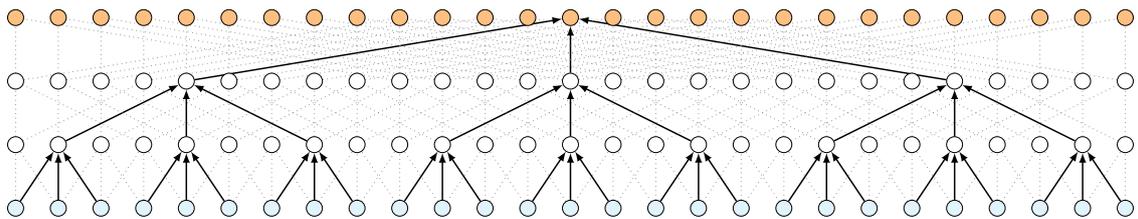


Figure 2: A-TCN architecture (kernel size: 3, dilation factors: 1,3,9).

The problem of diacritics restoration makes it possible to utilize the self-supervised training paradigm. The grammatically correct sentences from the target language provide the annotated data, we just have to remove the diacritical marks to generate the input for the task.

We also considered the use-case when not all of the diacritical marks are missing. Instead of hard copying the characters with diacritical marks at inference, we decided to provide the model training examples where not all of the marks are removed. The used augmentation method is similar to BERT’s MLM training technique [4]. In each epoch we removed a different random 80% of the diacritical marks.

3 Experimental Setup

For training we used a subset of the Hungarian Webcorpus 2.0 [8], a large collection of Hungarian texts from the Common Crawl and the Hungarian Wikipedia. We trained our models on 119,660 documents, overall containing 363 million characters. The validation dataset contained 14,958 documents, overall 45 million characters long. The texts were randomly sampled from the ”2019” part of the Common Crawl subcorpus, before train-dev-test cut.

The details of the architecture and the hyperparameters of the final model are the following. The character embedding vectors are of size 16. After the embedding the vectors are upsampled to dimension 176, which is the channel size. Zero padding is used to ensure that the output is the same length as the input. The network contains 4 residual block layers with dilation factors of 1,2,4, and 8, respectively. Each block contains 2 convolutional layers, each followed by batch normalization, ReLU, and spatial dropout layers, respectively. The convolutions have a kernel size of 5. The dropout rate is set to 0.2.

4 Results

Our model can be converted to ONNX (Open Neural Network Exchange), a cross-platform neural network format. ONNX.js is a JavaScript library, which can run models in ONNX format, which makes it possible to run our model in the browser. The demo of our model is available at: <https://web.cs.elte.hu/~csbalint/diacritics/demo.html>.

Converting to ONNX.js is not trivial. For example LSTM models are not supported, and even 1D convolutions had to be simulated with 2D convolutions. Another difficulty is that the model allows arbitrary input lengths, but in ONNX.js the first inference fixes the input sequence length. The solution is to dynamically reload the model. If the input is longer than the current limit, the model is reloaded with double length.

For baseline we chose Hunaccent [1], a decision tree based diacritics restorator, because it shares our goal to implement a small footprint restorator. Moreover, our solution can be run locally in a browser. To ensure a fair comparison, we set up our model to have a size similar to the 12.1 MB of the trained model of Hunaccent. The raw ONNX file of our trained model is 9.5 MB. Our demo HTML file is 12.72 MB. The HTML file contains the ONNX file as a Base64 encoded string.

Compared to the baseline, our model achieved significantly better results in all of the metrics we considered. *Character* accuracy measures the ratio of the correct characters in the output. *Important character* accuracy is measured by characters for which diacritical marks are applicable. In the case of the Hungarian language, these characters are the vowels. *Crude word* accuracy is measured by the ratio of the correct words in the output, where the words are defined in the simplest way by splitting the text along spaces.

	Character	Important character	Crude word
hunaccent	98.39	95.16	89.25
A-TCN	99.67	99.01	97.66

Table 1: Accuracy comparison between the baseline and our model

5 Acknowledgments

The research was partially supported by the Ministry of Innovation and Technology NRDI Office within the framework of the Artificial Intelligence National Laboratory Program, the Hungarian National Excellence Grant 2018-1.2.1-NKP-00008 and the grant EFOP-3.6.3-VEKOP-16-2017-00002.

The second author was supported by project "Application Domain Specific Highly Reliable IT Solutions" implemented with the support provided from the National Research, Development and Innovation Fund of Hungary, financed under the Thematic Excellence Programme TKP2020-NKA-06 (National Challenges Subprogramme) funding scheme.

References

- [1] **Ács, J. and Halmi, J.**, Hunaccent: Small Footprint Diacritic Restoration for Social Media, *Normalisation and Analysis of Social Media Texts (NormSoMe) Workshop*, 2016.
- [2] **Alqahtani, S., Mishra, A. and Diab, M.**, Efficient Convolutional Neural Networks for Diacritic Restoration, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, Hong Kong, 2019, 1442–1448.
- [3] **Bai, S., Kolter, J. Z. and Koltun, V.**, An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling, *arXiv preprint arXiv:1803.01271*, 2018.
- [4] **Devlin, J., Chang, M.-W., Lee, Kenton and Toutanova, K.**, BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, *arXiv preprint arXiv:1810.04805*, 2018.
- [5] **He, K., Zhang, X., Ren, S. and Sun, J.**, Deep residual learning for image recognition, *In Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, 770–778.
- [6] **Laki, L.J. and Yang, Z.G.**, Automatic Diacritic Restoration With Transformer Model Based Neural Machine Translation for East-Central European Languages, In: *ICAI*, 2020, 190–202.
- [7] **Náplava, Jakub and Straka, Milan and Straňák, Pavel and Hajic, Jan**, Diacritics restoration using neural networks, *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, 2018
- [8] **Nemeskey, D.M.**, *Natural Language Processing Methods for Language Modeling* (PhD thesis), Eötvös Loránd University, Budapest, 2020.
- [9] **Open Neural Network Exchange (ONNX)**, <https://github.com/onnx/onnx>
- [10] **ONNX.js**, <https://github.com/microsoft/onnxjs>
- [11] **Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. and Salakhutdinov, R.**, Dropout: a simple way to prevent neural networks from overfitting, *The journal of machine learning research*, 2014, 1929–1958.
- [12] **van den Oord, A. et al.**, WaveNet: A Generative Model for Raw Audio, *arXiv preprint arXiv:1609.03499*, 2016.
- [13] **Yu, F. and Koltun, V.**, Multi-scale context aggregation by dilated convolutions, *arXiv preprint arXiv:1511.07122*, 2015.

Nucleus classification with neural networks

Gábor Hidy András Lukács

Eötvös Loránd University, Institute of Mathematics
Department of Computer Science, AI Research Group
hidygabor@student.elte.hu, lukacs@cs.elte.hu

Abstract

Medical image recognition, e.g. that of histopathological images, provides unique challenges compared to more traditional computer vision tasks. One such problem is the scarcity of properly annotated, publicly available data. One frequent solution to insufficient data size is the use of transfer learning, where the network is pre-trained on a different, larger dataset from a similar domain, so it learns to extract low-level features from a wider array of data points.

Different variants of a ResNet model have been tested on a dataset of 24 000 nuclei sorted into seven different classes. The goal is to try to see what level of accuracy the model is capable of with this task, while trying out different practices used in standard image classification, like data augmentation and the use of weights pre-trained on ImageNet.

1 Dataset

The CoNSeP – ‘colorectal nuclear segmentation and phenotypes’ – dataset introduced in [3] is a dataset consisting of 41 images extracted from whole slide images that were taken from colorectal adenocarcinoma patients, with haematoxylin and eosin staining technology, at 40× magnification. Each image has a size of 1000 × 1000 pixels and is stored in PNG format. The images come from a total of 16 patients, and each contains at least a few hundred, and, in some cases, up to a thousand nuclei.

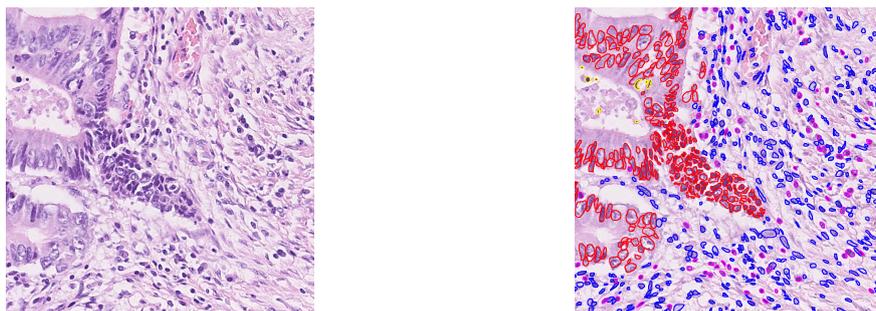


Figure 1: An example from the CoNSeP dataset, with the nuclei marked on the right [3]

Each nucleus is annotated according to a consensus of two expert pathologists in a pixel-wise manner. The nuclei had also been divided into seven classes: malignant/dysplastic epithelial, normal epithelial, fibroblast, inflammatory, muscle, endothelial, or miscellaneous. Each nucleus is labelled as one of the above classes, and each belongs only to one class, based on which type of cell it originates from.

The CoNSeP dataset in itself is not fit for classification, since each of its entries contains nuclei of multiple classes. Therefore a derived dataset needed to be made for classification. For this, each nucleus was extracted individually. A best-fitting square bounding box, with sides parallel to the horizontal and vertical axes, was placed on each nucleus, and then two pixels were added to each side. The obtained square images were then resized to 32×32 pixels, with the few below 10×10 pixels being discarded. The 32×32 size was chosen because that represents a median of the original size of the nuclei.



Figure 2: An example of each of the seven classes in the derived dataset

2 Experiments

The data was preprocessed with an RGB to BGR conversion, and the mean intensity of each channel of each image shifted to match that of the ImageNet dataset [2]. During training, images were padded with 4 black pixels on each side, and then a 32×32 sample was randomly cropped from the padded image or its vertical flip, and the resulting crop was then rotated with either 0° , 90° , 180° , or 270° . This augmentation technique is similar to that of [4].

One question that arises in any image classification task is how to measure the performance of the model. The usual practice is that the model is only trained on a portion of the dataset – usually around 75-80% –, and the rest is used as validation and test data. This is made complicated by an inhomogeneous dataset such as this one: nuclei coming from the same slides and the same patients will be much more similar to each other than ones from different slides and patients. To account for this discrepancy, two different train-validation splits have been applied: either 20% of data was split off for validation randomly, or all nuclei coming from 5 randomly selected images were used as validation.

3 Results

A ResNet-50 model [4] was used for experiments, with SGD optimisation. The models and experiments have all been implemented using the Keras API. [1] As anticipated, results largely differed depending on the validation datasets. Figure 3 shows the two extremes in validation accuracy, when training and validating on different subsets of the data. Both validation sets consist of nuclei coming from five different images, the difference lies in the choice of those five. With a particularly unfortunate choice, the model’s performance on the validation set fails to improve after the first few epochs, while another split produces much more promising results.

All experiments show that with the usual 10^{-1} or 10^{-2} learning rates, the model fails to converge, so a starting learning rate of 10^{-3} was used. Transfer learning and

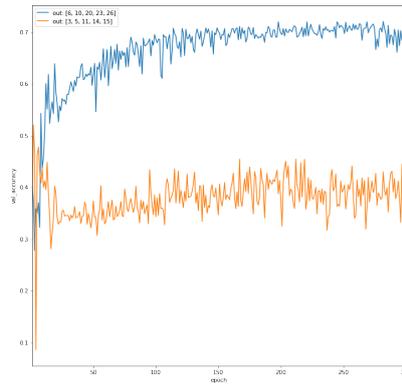


Figure 3: Comparing validation accuracy measured with two different train-validation splits

	5-out validation			Random split validation		
Method	Split 1	Split 2	Split 3	Split 4	Split 5	Split 6
Randomly initialised weights						
Without dropout	41.93%	68.44%	53.67%	65.27%	70.23%	67.18%
Pre-trained weights						
Without dropout	43.34%	72.81%	53.78%	69.42%	70.04%	68.93%
With dropout	49.48%	69.54%	54.95%	69.36%	71.48%	69.88%

Table 1: Validation accuracies on different validation sets

fine tuning with weights pre-trained on ImageNet have also shown to improve accuracy, compared to randomly initialised weights. Figure 4 illustrates the difference between the two initialisation techniques: the pre-trained model reaches higher accuracy faster than its counterpart.

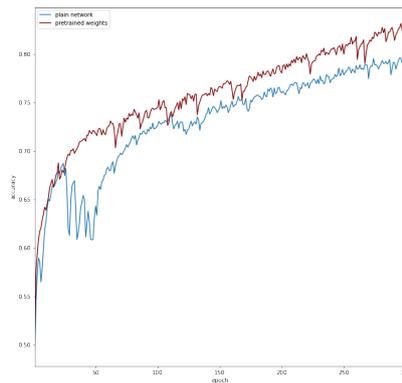


Figure 4: Comparison of the training accuracies of models with pre-trained and randomly initialised weights

Using dropout [5] before the classification layer of the network also seems to improve validation accuracy by anywhere between 1% and 5%, depending on the validation set. Table 1 compares the validation accuracy of different methods, with the model trained

and validated on different splits. Splits 1–3 validate on nuclei coming from 5 images, and train on the rest, while 4–6 splits the dataset into train and validation data randomly.

As seen, one of the main difficulties comes from the substantially differing accuracies measured on different train-validation splits. While one of the main sources of these differences is the fact that cells coming from the same person will be more similar to each other, another cause might be that when the pictures were taken, different amounts of haematoxylin and eosin were used for different samples, and the model learned faulty associations as a consequence of that. In the future, colour augmentation can be used to try to mitigate this effect.

Medical image recognition has its own challenges and tricks. Image classification of this sort of data proves to be a more difficult test than a similarly sized, more traditional image dataset. We have provided insight into an ongoing project, shown some of its current shortcomings, and some possible solutions to those. Our hope is that by obtaining more training data, by carefully calibrating preprocessing, and perhaps by choosing models more fit for this task, the initial results here can be meaningfully improved upon.

4 Acknowledgements

The research was partially supported by the Ministry of Innovation and Technology NRDI Office within the framework of the Artificial Intelligence National Laboratory Program, the Hungarian National Excellence Grant 2018-1.2.1-NKP-00008 and the grant EFOP-3.6.3-VEKOP-16-2017-00002. The second author was supported by project "Application Domain Specific Highly Reliable IT Solutions" implemented with the support provided from the National Research, Development and Innovation Fund of Hungary, financed under the Thematic Excellence Programme TKP2020-NKA-06 (National Challenges Sub-programme) funding scheme.

References

- [1] **Chollet, F. et al.** Keras, 2015, <https://keras.io>
- [2] **Deng, J., Dong, W., Socher, R., Li, L., Li, K., and Fei-Fei, L.**, ImageNet: A large-scale hierarchical image database, in: *CVPR 2009* (Miami, 2009), Conference on Computer Vision and Pattern Recognition, IEEE, 2009, 248–255.
- [3] **Graham, S., Vu, Q. D., Raza, S. E. A., Azam, A., Tsang, Y-W., Kwak, J. T., and Rajpoot, N.**, HoVer-Net: Simultaneous segmentation and classification of nuclei in multi-tissue histology images, *Medical Image Analysis*, **58** (2019).
- [4] **He, K., Zhang, X., Ren, S., and Sun, J.**, Deep residual learning for image recognition, in: *CVPR 2016* (Las Vegas, 2016), Conference on Computer Vision and Pattern Recognition, IEEE, 2016, 770–778.
- [5] **Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R.**, Dropout: A simple way to prevent neural networks from overfitting, *Journal of Machine Learning Research*, **15** (2014), 1929–1958.

Transfer learning for medical image classification

Gellért Károlyi András Lukács

Eötvös Loránd University, Institute of Mathematics
Department of Computer Science, AI Research Group
karolyigele@gmail.com, lukacs@cs.elte.hu

Abstract

While solving computer vision problems with deep learning, transfer learning is often used. The pretraining is usually done on ImageNet or another large dataset of natural images, but it is unclear how well features learned in this setting transfer to images from a completely different domain: medical images. First we confirmed that pretraining always has a positive effect on validation accuracy. We then found that, surprisingly, the effect of pretraining does not increase as we decrease the size of the training dataset. We show that pretraining helps less below a threshold in the low data regime and the beneficial effects of pretraining do not diminish even with tens of thousands of training images.

1 Introduction

In deep learning computer vision problems, Convolutional Neural Networks (CNNs) are most commonly used. Pretraining these CNN models on a large dataset often speeds up convergence during training, and models also tend to converge to better local minima. The pretraining is usually done on ImageNet, however, in the literature, there is conflicting evidence on how transferable these weights are to semantically different datasets like medical images. Medical images are generally grayscale, while natural images are colored and models usually have to differentiate between only about 2 to 20 categories as opposed to ImageNet's 1000. Medical images also tend to have less interclass diversity. Furthermore important parts of medical images can often take up a tiny area of the image, so attention to detail or texture may be more important. Despite these differences in image distribution there is evidence [4] that pretraining can boost performance. We confirm this with several commonly used models with different architectures. We then investigate how these models perform with different amounts of training data, specifically looking to confirm the benefits of pretraining in the low data regime, as medical image datasets often only contain a few hundred to a few thousand images.

2 Experimental setup

For our experiments we used the publicly available CheXpert [3] dataset, which consists of 224,316 chest radiographs with up to 14 labels each. There are images in both frontal and lateral views. We only used frontal images, because frontal and lateral images differ so much that if we used both, the models would essentially have to learn 2 different tasks simultaneously; and there are more frontal images. The task we trained on was to learn only one of the labels, namely Lung Opacity. We chose lung opacity, because it should be detectable visually from chest x-rays and because there were a lot of labels for

it. There were 94,211 positively and 5,051 negatively labeled images, along with 16,974 images of healthy patients. Combining normal images with the images negative for lung opacity we got 20,170 images. To make a balanced dataset we used 19,000 negative + 19,000 positive = 38,000 images for learning, while reserving 2,000 images for validation.

When using training sets with sizes less than 38,000, we repeated the training set to make training epochs (somewhat) comparable for vastly different training set sizes. Since we shuffled the datasets before each epoch, this may mean that each individual image can appear more than once in a batch. Extending epochs this way was mainly to facilitate consistent early stopping and for the ability to meaningfully compare epochs between runs.

Sabottke and Spiele [5] found that for chest radiographs, if the image resolution is less than 224 by 224, then the information loss due to compression is significant, however above 256 by 256 pixels, accuracy seems to plateau. We wanted to make sure our image resolution was not too far from the resolution of the ImageNet images the models were originally trained on (224 X 224 X 3). For these reasons we chose an image size of (256 X 256 X 3). Chest radiograph images are grayscale, but in order to use the pretrained weights, we duplicated the single color channel. Batch size was chosen to be 64.

We used Coolmomentum [1] optimizer with $\rho_0 = 0.99$, $\alpha = 0.99997$ and a starting learning rate of 0.0001. We decreased the learning rate exponentially by a total factor of (at most) 100 over a maximum of 100 epochs, using early stopping when training accuracy did not improve by at least 1% for 5 consecutive epochs. This was only to automatically stop training after model convergence, as this time varied based on the size of the training set and pretraining (or the lack thereof). This meant that for the larger training sets training took longer and also reached a smaller learning rate. Because of this, overfitting was more prevalent in experiments with larger training sizes. We recorded the validation accuracy after every epoch, and in Section 3, we report the values we would have gotten stopping at the highest epoch validation accuracy.

3 Experiments

We were interested in how two hyperparameters impacted the transferability of the weights: model architecture and training set size. For weight transfer, we used the fine tuning approach. We removed the dense top layers of each of the models and added a new linear classifier with a single output neuron. We always initialized the classification layer randomly while initializing the rest of the model either randomly, or from the ImageNet pretrained weights. We tried several well-known models from different stages of the evolution of CNNs, but with parameter counts in the same order of magnitude: VGG16 [6], ResNet50 [2], and InceptionV3 [7]. Ke et al. [4] found that model size does not impact performance as significantly as model family, this is why we only tried one model per model family.

According to industry wisdom, transfer learning is the most useful when the training size is too small to train from scratch. We wanted to test this theory by limiting training to only a small subset of the training set of sizes 200, 2000, 8000, and 38000. We ran each setup 10 times to reduce the effect of randomness from selecting a random subset of images and other inherent sources of randomness that occur during training.

For augmentation we used horizontal flipping for all the experiments. For the training

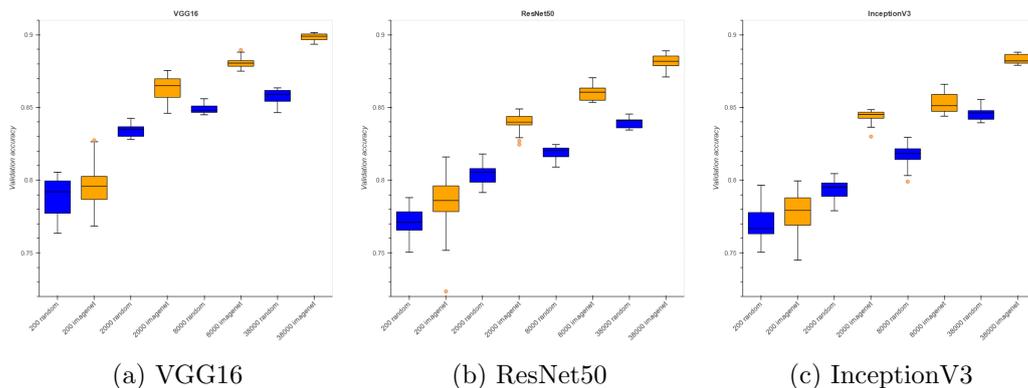


Figure 1: Validation accuracy results for the different models. The orange boxes correspond to experiments with ImageNet initialization, while blue boxes correspond to experiments with random initialization. All experiments were repeated a total of 10 times.

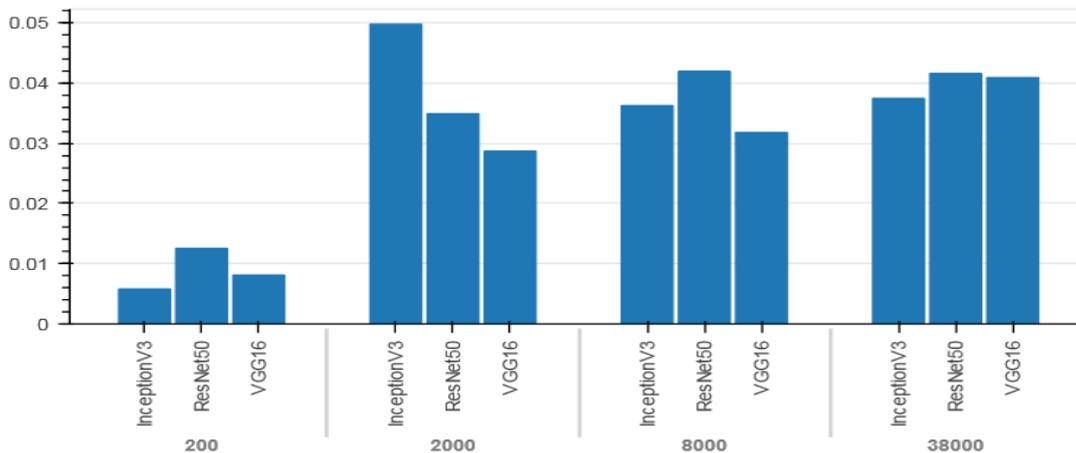


Figure 2: Boost in average validation accuracy from pretraining

set size 200 experiments, we also changed the brightness by up to 0.2. For the size 2000 experiments, we used random rotation by at most 0.1π .

Results are illustrated in figure1, where each plot shows results from a single model and each box corresponds to a set of 10 experiments with the same setup with training set size and initialization varying between them.

4 Conclusion

Our experiments show that, unsurprisingly, models trained on bigger datasets always performed better, but interestingly the oldest VGG16 model slightly outperformed both ResNet50 and InceptionV3, suggesting that newer model architectures may have started overfitting to ImageNet.

Surprisingly we found that transfer learning may not be the most useful in the low data regime. Despite still being able to somewhat learn the task, the difference between

pretrained and random weights was clearly the least significant in the case of the smallest training dataset. On the other hand, pretraining seems to be just as important with tens of thousands of training images, suggesting that medical image tasks may need at least hundreds of thousands of images before the importance of pretraining might fall off. Overall we can conclude that pretrained weights are always worth using, when they are available, but with a very small training set (under about 1000 images) custom model architectures may be more beneficial.

5 Acknowledgments

The research was partially supported by the Ministry of Innovation and Technology NRDI Office within the framework of the Artificial Intelligence National Laboratory Program, the Hungarian National Excellence Grant 2018-1.2.1-NKP-00008 and the grant EFOP-3.6.3-VEKOP-16-2017-00002. The second author was supported by project "Application Domain Specific Highly Reliable IT Solutions" implemented with the support provided from the National Research, Development and Innovation Fund of Hungary, financed under the Thematic Excellence Programme TKP2020-NKA-06 (National Challenges Sub-programme) funding scheme.

References

- [1] **Borysenko, O. and Byshkin, M.**, CoolMomentum: A Method for Stochastic Optimization by Langevin Dynamics with Simulated Annealing, *arXiv preprint arXiv:2005.14605*, (2020)
- [2] **He, K., Zhang, X., Ren, S and Sun J.**, Deep residual learning for image recognition, *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, 770–778
- [3] **Irvin, J. and Rajpurkar, P. et al.**, Chexpert: A large chest radiograph dataset with uncertainty labels and expert comparison, *Proceedings of the AAAI Conference on Artificial Intelligence* **33** (01) 2019, 590–597
- [4] **Ke, A, Ellsworth, W., Banerjee, O., Ng, A. Y. and Rajpurkar P.**, CheX-transfer: performance and parameter efficiency of ImageNet models for chest X-Ray interpretation, *arXiv preprint, arXiv:2101.06871* (2021)
- [5] **Sabottke, C. F. and Spieler, B. M.**, The effect of image resolution on deep learning in radiography, *Radiology: Artificial Intelligence* **2** (1) e190015, (2020)
- [6] **Simonyan, K. and Zisserman, A.**, Very deep convolutional networks for large-scale image recognition, *arXiv preprint arXiv:1409.1556* (2014)
- [7] **Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J. and Wojna, Z.**, Rethinking the inception architecture for computer vision, *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, 2818–2826

Global Sinkhorn Autoencoder - Optimal transport on the latent representation of the full dataset

Melinda Kiss¹, Adrián Csiszárík^{1,2}, Ákos Matszangosz², Balázs Maga¹, and Dániel Varga²

¹*Eötvös Loránd University, Budapest, Hungary*

²*Alfréd Rényi Institute of Mathematics, Budapest, Hungary*

melindafkiss@gmail.com, csadrian@renyi.hu, matszangosz.akos@renyi.hu,
mbalazs0701@gmail.com, varga.daniel@renyi.hu

Our objects of interest are generative autoencoders, that is, autoencoders where the decoder can be used as a generator when samples from some prior distribution are fed to it. More specifically, we are interested in the Wasserstein Autoencoder (WAE) class of deep learning models [1]. Tolstikhin et al’s original formulation states that if the pushforward of the true data distribution is equal to the prior, then the Wasserstein distance between the true data distribution and the generated distribution is equal to the reconstruction error. Thus, by applying Lagrangian relaxation, the loss function for these models is the reconstruction error plus a penalty term for matching the aggregated posterior to the prior. This penalty term can be seen as a statistical test verifying that the pushforward is indeed close to the prior.

In all incarnations of this idea that we are aware of, such as Adversarial Autoencoders [2], WAE-MMD [1], MMD nets [3], or Sinkhorn Autoencoders [4], the input for this statistical test is the latent image of the minibatch (latent minibatch for short). However, as it is already hinted by experiments published by Rubinstein et al. [5], the size of the minibatch may strongly affect the performance of the model.

We argue that in the WAE class of models, the minibatch size (which is typically in the range of 50-200), is not large enough compared to the dimension of the latent space, which means that the statistical test is too weak to guarantee a good match between the pushforward and the prior distribution. We propose an optimal transport-based generative model from the Wasserstein Autoencoder family of models, with the following innovative property: the optimization of the latent point positions takes place over the full training dataset rather than over a minibatch. Our baseline model is the Sinkhorn Autoencoder [4], which operates on the latent embedding of the mini-batch. We compare our *Global Sinkhorn Autoencoder* model with the “local” Sinkhorn Autoencoder baseline on natural and synthetic datasets, on several evaluation metrics.

Detailed description of the Global-SAE algorithm. Our main proposal for new methods capable to work with point cloud sizes strongly exceeding the ones encountered in

Algorithm 1 GLOBAL SINKHORN AUTOENCODER (GLOBAL-SAE)

Input: Dataset $x = \{x_i\}_{i=1}^n$, prior distribution P_Z ,
encoder weights Ψ , decoder weights Φ , learning rate μ , training iters $iter$, minibatch size M
parameters for the Sinkhorn algorithm: $\varepsilon \in \mathbb{R}$, $L \in \mathbb{N}$, $c = \|\cdot\|_2^2$, sinkhorn weight λ

Output: Trained model with encoder weights Ψ , decoder weights Φ

- 1: **for** $i = 1..iters$ **do**
- 2: $z = \{z_i\}_{i=1}^n \sim P_Z$ {Sample target point set from P_Z — global data}
- 3: $\hat{z} = \{\hat{z}_i\}_{i=1}^n \leftarrow Q_\Psi(x)$ {Encoded image of the entire dataset x — global data}
- 4: $\tilde{x} = \{x_i\}_{i \in I}$ where $I \subseteq \{1, 2, \dots, n\}$ random subset, $|I| = M$ {Minibatch sample from x }
- 5: $\tilde{x}' \leftarrow G_\Phi(Q_\Psi(\tilde{x}))$
- 6: $D \leftarrow \frac{1}{M} \|\tilde{x} - \tilde{x}'\|_2^2$ {Calculate reconstruction loss for minibatch}
- 7: $S = S_{c,\varepsilon,L}(\hat{z}, z)$ {Calculate Sinkhorn loss for global point clouds \hat{z} and z — global data}
- 8: $(\Psi, \Phi) \leftarrow (\Psi, \Phi) - \mu \cdot \nabla_{(\Psi, \Phi)}(D + \lambda \cdot S)$ {Update model parameters}
- 9: **end for**

the minibatch regime is summarized in Algorithm 1, in which we proceed as follows. In lines 2-3 we resample the target latent point cloud, and calculate the latent images \hat{z} of the dataset x . Note, that instead of considering data and target samples of size M of a minibatch, we examine the complete dataset $x = \{x_i\}_{i=1}^n$ and a correspondingly sized target set $z = \{z_i\}_{i=1}^n$ sampled from the prior distribution P_Z , thus we operate in the global scope of the dataset. In lines 4-6 we calculate the reconstruction loss for a minibatch. (For this regular autoencoder loss term we remain with the minibatch scope.) In line 7 we calculate the error terms resulting from the optimal transport cost between \hat{z} and z . Then in line 8 we update the model parameters by taking a gradient step with the above global optimal transport loss function and also with the reconstruction error.

Calculate in the global scope, but backpropagate only on a minibatch. In the Global-SAE algorithm all latent positions are calculated by the encoder in each iteration, thus the gradient signal coming from the Sinkhorn loss can be backpropagated for all datapoints to the encoder weights. In contrast to this, we introduce a new variant: Minibatch-Global-SAE, which takes the latent positions from a "cache" for all points except for the current minibatch. This means that for this variant, the gradient signal coming from the Sinkhorn loss can only propagate to the encoder for the elements of the current minibatch, and the gradient signal is "thrown away" for all datapoints outside the minibatch. The only difference between the two algorithms is that in case of Minibatch-Global-SAE we take the gradient on the minibatch, while in Global-SAE we take the gradient on the entire dataset. The runtime and the memory requirements for the decoder forward-backward pass and the Sinkhorn loss calculation are identical between the two variants. The significant differences between the resource requirements of the two variants lie in the way encoder gradient updates are treated. The Full Global variant requires a forward-backward pass of the encoder on the full dataset for each minibatch calculation.

The MNIST dataset can be seen as a balanced mixture of 10 disjoint image datasets,

one for each digit. Thus it is reasonable to consider a 10-mixture of Gaussians as the latent prior distribution for such a dataset, as was already done by [2]. (Setting the pairwise KL-divergences of the Gaussians to a high value, reflecting the fact that the class ambiguity of the data distribution is small.) In this setup, a goal we might set for our encoder is that the pushforward of the data distribution of a single image class should be close to one of the Gaussians. Conversely, the decoder pushforward of a single Gaussian should be close to the data distribution of a single image class.

To obtain a detailed view on the quality of the models, we examine five different quantitative evaluation metrics, each of which shed light from a different viewpoint on the quality of the learned model. Besides reporting the standard *test reconstruction loss*, and the Sinkhorn loss on the entire test set (which we call *global OT loss* for brevity), we also introduce three metrics that measure the quality of the latent space formed by the trained model.

Local clustering To measure how well the same labeled points coalesce we utilize the *local clustering* metric which is a standard evaluation metric in semisupervised models. Here, we call an encoded point good, if the 10 nearest encoded points to it in the latent space have the same label as this point. The ratio of the good points in the test set is what we call *local clustering*.

Cluster matching. One might wish that working with such a prior (the "flower" prior), as there are 10 labels, all encoded images with the same label should belong to the same petal. We are interested in how much the actual embedding approximates this ideal. This consideration leads to a metric we call *cluster matching*. Informally, the metric is the classification performance of the clustering algorithm assigning the points to petals by maximum likelihood, assuming that the unknown assignment between clusters and labels is chosen optimally. First, we split the plane by 5 lines passing through the origin, such that each angle between adjacent lines is the same, and for each angular domain, the angular bisector passes through the mean of one of the Gaussian distribution in the mixture. We partition the plane into 10 domains, and we can assign each encoded point to the petal which has its mean in that specific domain. Then we create a complete bipartite graph where the two sets of nodes represent the labels and the petals respectively, and the weight of the edge between the i th petal and j th label is equal to the number of test points assigned to the i th petal and has label j . The value *cluster matching* is the weight of the maximal weight perfect matching divided by the size of the test set.

Covered area. As the latent space is two dimensional, we can check visually how well the models are able to match the encoded points to the prior point cloud. In addition we worked out a measure which we call *covered area* which checks how well the encoded points are matched to the prior distribution.

For Gaussian priors, let T be a transformation that transforms the prior to the uniform distribution on the unit square in the plane. We transform each encoded point by T , thus each encoded point is transformed to the unit square. We create a very dense grid on the square, look at the small neighborhood of the transformed encoded points and compute how many little squares in the grid intersect with the neighborhoods. The ratio of such squares to the number of squares in the grid is what we call *covered area*.

For Gaussian mixtures, we assign each encoded point to one of the mixture components

by maximal likelihood. Assuming large divergence between our mixture components, every such “petal” is a Gaussian distribution with good approximation, and we can calculate the covered area for each of the petals separately, then take the average of these values.

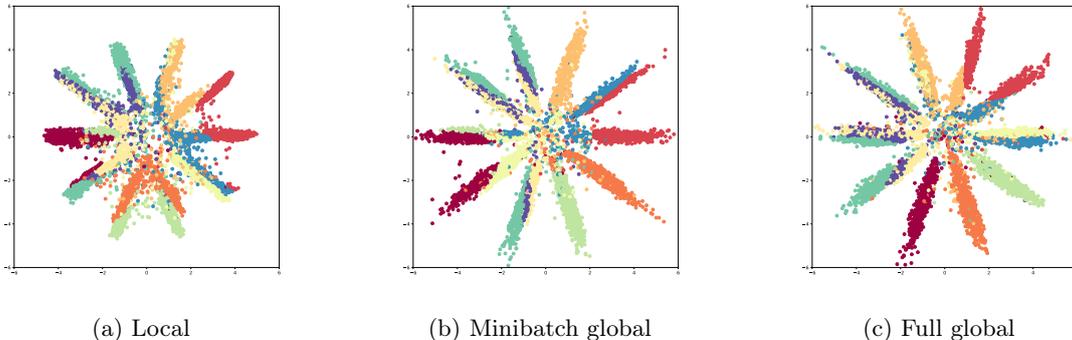


Figure 1: The encoded test set in the latent space, using an MLP net. The points match the prior better and they have a more orderly arrangement in the global versions.

MLP	Local	Minibatch global	Full global
Sinkhorn loss ↓	0.018 ±0.28e−2	0.013 ±0.33e−2	0.014 ±0.38e−2
Reconstruction ↓	0.036 ±0.19e−3	0.034 ±0.33e−3	0.034 ±0.1e−3
Local clustering ↑	0.456 ±0.012	0.505 ±0.019	0.516 ±0.017
Cluster matching ↑	0.55 ±0.067	0.64 ±0.033	0.64 ±0.01
Covered area ↑	0.83 ±0.82e−2	0.89 ±0.007	0.87 ±0.008

Table 1: Results for the MLP net after 50 epochs. Averages of 5 runs with different random seeds. An arrow indicates if lower (↓) or higher (↑) is better.

As our experiments demonstrate, our global models consistently improve on the local baselines for complex priors in low latent dimensions.

References

- [1] Tolstikhin, I. and Bousquet, O. and Gelly, S. and Schoelkopf, B.: Wasserstein Auto-Encoders, *ICLR*, 2018.
- [2] Alireza Makhzani and Jonathon Shlens and Navdeep Jaitly and Ian Goodfellow: Adversarial Autoencoders, International Conference on Learning Representations, <http://arxiv.org/abs/1511.05644>, 2016.
- [3] Dziugaite, G. K. and Roy, D. M. and Ghahramani, Z.: Training generative neural networks via Maximum Mean Discrepancy optimization, *AUAI*, 2015.
- [4] Patrini, G. and Carioni, M. and Forre, P. and Bhargav, S. and Welling, M. and den Berg, R. and Genewein, T. and Nielsen, F.: Sinkhorn autoencoders, arXiv preprint arXiv:1810.01118, 2018.
- [5] Rubenstein, P. K. and Schoelkopf, B. and Tolstikhin, I.: Wasserstein Auto-Encoders: Latent Dimensionality and Random Encoders, *ICLR* workshop, 2018.

Machine Learning Algorithms for MOD Lapse at renewal

Péter Marton^{1,2} Norbert Bicskei¹ András Lukács²

¹Allianz Hungary

²Eötvös Loránd University, Institute of Mathematics, AI Research Group

`peter.1.marton@allianz.hu`, `norbert.bicskei@allianz.hu`, `lukacs@cs.elte.hu`

Abstract

In this paper we are focusing on the modelling of Motor Own Damage insurances' cancellation by client at annual. While in the insurance company Generalized Linear Model (GLM) is used mainly for modelling purposes, it lacks the ability to effectively identify non-linear interactions. To increase the predictiveness of our models, we experimented with different hyper-parameters for Random Forest and Gradient Boosting Regression, used their results both directly as the expected value for the insurance policy cancellation and using a threshold to decide will the client actually cancel or not the policy at anniversary. We have analysed the obsolescence of data, to improve our practice, which using fixed number of previous years to build the training set. For pricing, the most important to predict sum of actual lapse on sub-portfolios which are part of either the tariff premium or the actual risk (so called technical premium) or both, thus we concentrated on the global actual vs expected value, and analysed them on different sub-portfolio using an interactive excel one-way analysis tool. This brought good results, thus we also started researching good models to predict policy-level lapse probability, which could support our ongoing product modernization campaign called to life partially by the covid-situation. Finding a technique to predict for individual outcomes even in million rows database made necessary using more advanced machine learning algorithms.

1 Introduction

In this paper we present results given during building a model to predict the probability of policies of the actual MOD portfolio to be cancelled at anniversary by client. This is a behavioural model which harder to explain with the available (mainly tariff-related thus risk-explanator) parameters, as it depends clients' discretionary decision and competitors' agents' and brokers' activity, triggered by not necessary reasonable decisions. The ascertainments below shows toward a machine learning method to predict cancellations better than the current industry-used GLM methods.

2 Overview of data by heat maps

There was a tool introduced to get heat maps directly from the Oracle database (and store it in Oracle tables in a BLOB as a 32 bit transparency supported windows bitmap). This can produce an X-Y coordinate system (where either the X or the Y or both can be numeric or categorical variable), and visualize two numeric value (like lapse ratio, and a so called exposure which means the amount of data, where the lapse ratio comes from). This 2 value allows us the highlight the most important parts, and reduce the noise by fade regions on the map where extreme ratios would come from tiny exposure and would jiggle

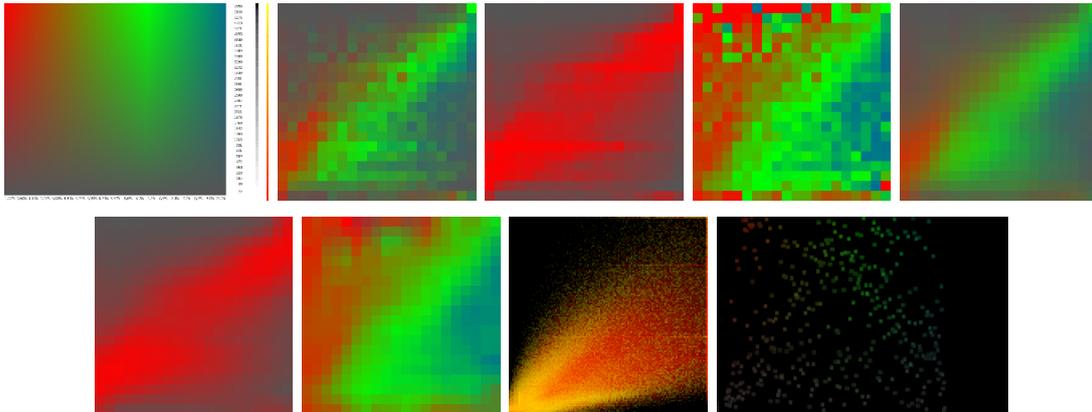


Figure 1: Heat maps with native and blurred results by saturation and luminosity weight visualisation, actually used weight-values

the map. The main value (like lapse ratio) could be the hue, going from red to either yellow or green or blue. The weight (the exposure) used to be either the brightness (black is 0 exposure) or the saturation (gray is the zero exposure) or the transparency (completely transparent is the zero exposure), either these two legend combined into a rectangle or visualised separately (Figure 1).

Using lumiance (brightness) for weighting would direct one’s attention more likely to the weighting than the value, therefore this is the less fortunate solution, as the weighting’s primary function to hide noise. Using to high resolution for heat map chart would also generate more noise and would make more difficult to catch the actual message of the chart. Using lower resolution causes only a smaller amount of the possible value/weight combination displayed actually which is also extracted with this tool, like actual picture down-sampled for better result in a small pdf embedded picture. Using transparency for weighting is beneficial when an actual geographical map is used as background, in this case -thanks to the precise geocoding roof – top addresses – we can produce virtually perfect resolutions for a whole country-sized map (Figure 2).

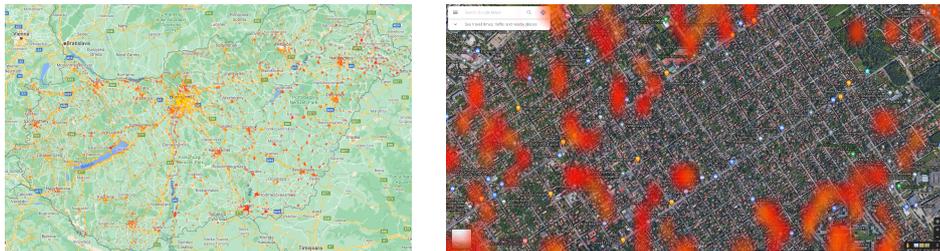


Figure 2: Arbitrary high resolution maps

3 Time Consistency

We have analysed the time consistency in the lapse modelling, as a single year used as a training set and a single as the testing set. These results are summarised in Table 1.

		TRAINING YEAR						
		2014	2015	2016	2017	2018	2019	2020
TESTING YEAR	2014	62%	61%	60%	59%	57%	58%	58%
	2015	60%	62%	62%	59%	55%	56%	56%
	2016	58%	61%	62%	58%	55%	56%	55%
	2017	58%	60%	59%	61%	58%	57%	58%
	2018	56%	56%	55%	57%	62%	59%	59%
	2019	53%	55%	55%	55%	56%	61%	57%
	2020	56%	56%	54%	57%	60%	55%	62%

Table 1: Model’s AUC training on one year and testing on an other

Percentage here are AUCs, and the model was Random Forest [1] with 1,000 estimator and with a maximal depth of 3. The scores output by the model used as expected values for that policies. Note that with less optimal hyper-parameterized model these scores can easily result negative numbers. For models like GLM [3], where averages of subsets of the training set’s are being estimated there is a common overcome for this is to use a link function (like logit function) to extend the $[0, 1]$ interval to $(-\infty, +\infty)$. One can observe the weak performance of the model even when tested on the same year where the model was trained. There is a still unexplained difference in the year of 2019, while in 2018 there was a new product launched which altered the nature of the portfolio’s MOD lapse, resulting years before 2018 somewhat obsoleted. As one can expect, the higher distance in time between the train and test set, the weaker explanation power observed. This raised the question, how far would we go back in time to get data, where is the point where further appends from the past would not improve the model. This depends on our goals: to predict individual policies to lapse or renewal (where the AUC should be increased) or would we like to make a portfolio sized prediction where average vs expected fits better.

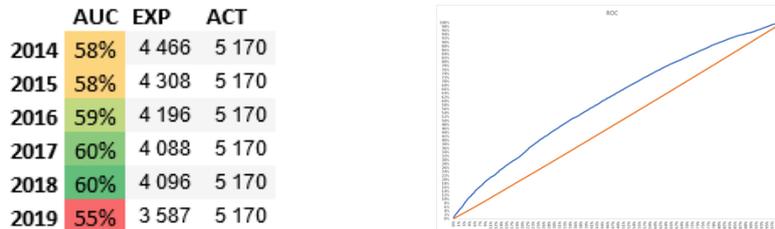


Figure 3: Ideal number of years to train depends on AUC/Actual vs expected optimisation

We found that 2 or maximally 3 years were optimal for AUC optimisation (which is exactly the point where the new product launched), while for a portfolio-sized estimation all the 6 tested previous years produced better explanations as they included to the test set (Figure 3). Actual implementation of the AUC-like measure would be beneficial for product migration where old policy’s policyholder are each-by-each looked up by the policy’s caregiver agent and the policyholder is offered to make a new contract with a modernised new product. On the other hand, for pricing the ideal approach is Towers Watson Emblem software’s point of view, where a whole sub-portfolio’s average is being estimated. This allows us to map probabilities to entire hernial and the actual tariff multipliers will be based on these measurement. That way, we have to optimise to that sub-portfolio’s actual vs expected values. To monitor that result, we developed an excel tool, to show these

values with fast clickable VBA supported charts (Figure 4).

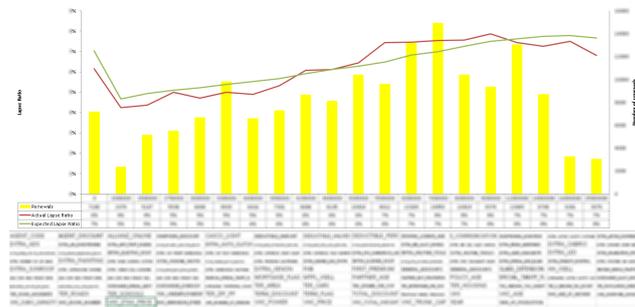


Figure 4: Actual vs expected monitored against 115 rating factors already

4 Hyper-parameter optimisation

There is a running project to implement thresholds to substitute ML method’s scores with 0/1 values and use this to estimate and analyse the results with confusion matrix, sensitivity and accuracy – yet no better results are achieved. This could let us optimize the hyper-parameterizing of the algorithms better. Hyper-parameter optimisation is a crucial point of machine learning modelling, we measured actual vs expected lapse on arbitrary selected parameters to implement modelling, and also generated some charts to recognise different algorithms behavior on the actual MOD portfolio. In general, Gradient Boosting Method [2] found to be more stable, which means less deviance is the test set, and reliable explanation of the grand mean on the training set (Figure 5).

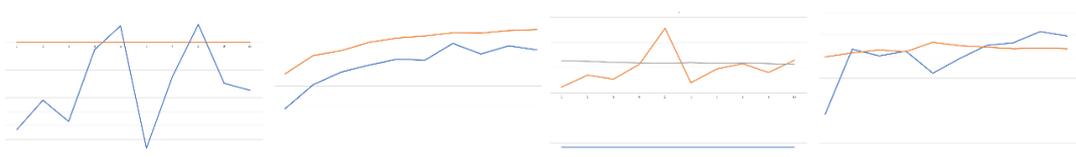


Figure 5: Optimisation of tree depth for Random Forest (blue) and GBM (red)

5 Acknowledgements

This work was prepared with the professional support of the Doctoral Student Scholarship Program of the Co-operative Doctoral Program of the Ministry of Innovation and Technology financed from the National Research, Development and Innovation Fund.

References

- [1] **Breiman, L.**, Random forests, *Machine learning*, **45**(1), 2001, 5-32.
- [2] **Friedman, J. H.**, Stochastic gradient boosting, *Computational statistics & data analysis*, **38**(4), 2002, 367-378.
- [3] **de Jong, P. and Heller, G. Z.**, *Generalized linear models for insurance data*, Cambridge University Press, Cambridge, 2008.

Section:

Type Theory

Organizer: Ambrus Kaposi

Invited talk:

Nicolai Kraus: Wellfounded and Extensional Ordinals in Homotopy Type Theory (talk on joint work with Fredrik Nordvall Forsberg and Chuangjie Xu)

Contributions:

- István Donkó and Ambrus Kaposi: Properties of Setoid Type Theory
- Ambrus Kaposi and Zongpu Xie: A model of type theory supporting quotient inductive-inductive types
- András Kovács: Staged Compilation and Generativity

Wellfounded and Extensional Ordinals in Homotopy Type Theory

Nicolai Kraus

(talk on joint work with Fredrik Nordvall
Forsberg and Chuangjie Xu)

University of Nottingham

`nicolai.kraus@nottingham.ac.uk`

Summary

In this talk, I discuss three different notions of ordinals in homotopy type theory and show how they relate to each other. The first are *extensional wellfounded orders*, the second are (a refined version of) *Brouwer trees*, and the third are *Cantor normal forms*.

This talk is based on recent joint work with Fredrik Nordvall Forsberg and Chuangjie Xu [6].

Introduction

Ordinals generalise the natural numbers, with examples being 0 , 27 , ω , or $\omega^\omega + 3$. The set of finite ordinals is just \mathbb{N} , and ω is the smallest ordinal that is larger than all natural numbers. Although ordinals can be infinite, decreasing sequences of ordinals are always finite; in other words, every decreasing sequence of ordinals terminates, a property known as *wellfoundedness*. This makes them suitable to show termination of processes or to justify induction principles. The usual set-theoretic definitions of ordinals further are *extensional*, i.e. if ordinals x and y are such that any z is smaller than x if and only if it is smaller than y , then x and y are equal.

How can the concept of ordinal numbers be represented in constructive type theory? If we consider a relation $<: X \rightarrow X \rightarrow \mathbf{Prop}$, then wellfoundedness can, as first suggested by Aczel [1], be formulated via an inductive predicate. More precisely, the predicate $\mathbf{Acc}: X \rightarrow \mathbf{Type}$ is defined inductively by a single constructor

$$\mathbf{acc} : (x : X) \rightarrow ((y : X) \rightarrow y < x \rightarrow \mathbf{Acc}(y) \rightarrow \mathbf{Acc}(x)), \quad (1)$$

and $<$ is *wellfounded* if every element is accessible. Further, $<$ is *extensional* if

$$(x y : X) \rightarrow (\forall z. z < x \leftrightarrow z < y) \rightarrow x = y. \quad (2)$$

We consider three notions of ordinals:

1. Extensional Wellfounded Orders

Translating the classical set-theoretic definition directly into a constructive setting leads to the following definition, as studied in the HoTT book [8, Chapter 10] and by Escardó [4]: An ordinal is a type X together with a relation $<: X \rightarrow X \rightarrow \mathbf{Prop}$ which is *transitive*, *extensional*, and *wellfounded*. We denote the type of all such ordinals by \mathbf{Ord} . It is known that \mathbf{Ord} is an ordinal itself, where the relation is given by *simulations* [8].

2. Brouwer Trees

In functional programming, *Brouwer (ordinal) trees* \mathbf{Brw} are often defined with the constructors `zero`, `successor` and a supremum constructor $\text{sup} : (\mathbb{N} \rightarrow \mathbf{Brw}) \rightarrow \mathbf{Brw}$. However, it would maybe be more accurate to think of elements of this type as representatives (or notations) for ordinals; for example, $\text{sup}(0, 1, 2, 3, \dots)$ and $\text{sup}(1, 2, 3, \dots)$ represent the same ordinal, but are of course different as elements of \mathbf{Brw} . In particular, \mathbf{Brw} is not extensional. To repair this defect, we turn \mathbf{Brw} into a quotient *inductive-inductive type* [2, 5] such that \mathbf{Brw} is defined simultaneously with its relation $<$. The relation $<$ of this version can then be shown to be extensional and wellfounded at the same time.

3. Cantor Normal Forms

In classical set theory, every ordinal α can be written in Cantor normal form, i.e. as

$$\alpha = \omega^{\beta_1} + \omega^{\beta_2} + \dots + \omega^{\beta_n} \quad (\beta_1 \geq \beta_2 \geq \dots \geq \beta_n). \quad (3)$$

If we consider binary unlabelled trees and write the node constructor $\text{node}(a, b)$ as $\omega^a + b$, then finite trees (with a condition corresponding to $\beta_1 \geq \beta_2 \geq \dots$) correspond to ordinals below ε_0 [3, 7]. They can, again, be equipped with an extensional and wellfounded relation.

Comparison

Although each of \mathbf{Ord} , \mathbf{Brw} , and \mathbf{Cnf} classically corresponds to a certain subset of the class of ordinal numbers, they behaved very differently in a constructive settings. Equality and the relation $<$ on \mathbf{Cnf} are decidable, while for \mathbf{Ord} , even deciding whether an ordinal is zero already corresponds to the law of excluded middle. \mathbf{Brw} sits in the middle: While our version makes it possible to decide whether a Brouwer tree is finite, and equality for finite ones is decidable, general equality is still undecidable. Moreover, a “more decidable” version can be embedded into a “less decidable” one, in the following sense:

After defining the typical arithmetic operations for \mathbf{Brw} , we have an obvious map $\text{CtoB} : \mathbf{Cnf} \rightarrow \mathbf{Brw}$, defined by $0 \mapsto 0$ and $\omega^a + b \mapsto \omega^{\text{CtoB}(a)} + \text{CtoB}(b)$. This map is injective and preserves and reflects both $<$ and \leq . More importantly, it commutes with $+$, $*$, and ω^x . It is also worth noting that CtoB is *bounded* by ε_0 which can be defined in \mathbf{Brw} but, of course, not in \mathbf{Cnf} .

Using that our order on \mathbf{Brw} is extensional and wellfounded, we have a second canonical map $\text{BtoO} : \mathbf{Brw} \rightarrow \mathbf{Ord}$, $a \mapsto \Sigma(y : \mathbf{Ord}).y < a$. This map is injective and preserves $<$ as well as \leq . It commutes with limits, but constructively not with successors. Assuming the law of excluded middle, it is a simulation (i.e. we have $\mathbf{Brw} < \mathbf{Ord}$), but this is not constructively provable. The map BtoO is bounded by \mathbf{Brw} .

Full paper and formalisation

A full paper with all details is available on the arXiv as [arxiv:2104.02549](https://arxiv.org/abs/2104.02549). We have formalised most of the results in cubical Agda (although without satisfying the termination checker in all cases), and this can be found at <https://bitbucket.org/nicolaikraus/constructive-ordinals-in-hott>.

References

- [1] **Aczel, P.**, *An introduction to inductive definitions*, Studies in Logic and the Foundations of Mathematics, volume 90, pages 739–782. Elsevier, 1977.
- [2] **Altenkirch, T., P. Capriotti, G. Dijkstra, N. Kraus, and F. Nordvall Forsberg**, *Quotient inductive-inductive types*, FoSSaCS'18, Springer, 2018.
- [3] **Buchholz, W.**, *Notation systems for infinitary derivations*, Archive for Mathematical Logic, 30:227–296, 1991.
- [4] **Escardó, M.**, *Agda implementation: Ordinals*, <https://www.cs.bham.ac.uk/~mhe/TypeTopology/Ordinals.html>, since 2010.
- [5] **Kaposi, A. and A. Kovács**, *Signatures and induction principles for higher inductive-inductive types*, Logical Methods in Computer Science, 16(1), 2020.
- [6] **Kraus, N., F. Nordvall Forsberg, and C. Xu**, *Connecting constructive notions of ordinals in homotopy type theory*, ArXiv e-prints, arXiv:2104.02549, 2021.
- [7] **Nordvall Forsberg, F., C. Xu, and N. Ghani**, *Three equivalent ordinal notation systems in cubical Agda*, CPP'20, pages 172–185. ACM, 2020.
- [8] **The Univalent Foundations Program**, *Homotopy Type Theory: Univalent Foundations of Mathematics*, <http://homotopytypetheory.org/book/>, Institute for Advanced Study, 2013.

Properties of Setoid Type Theory

István Donkó[†] Ambrus Kaposi[‡]

Eötvös Loránd University

Faculty of Informatics

Department of Programming Languages and Compilers

[†]ist1115@inf.elte.hu [‡]akaposi@inf.elte.hu

1 Motivation

Type theory provides a very useful toolkit for creating and validating formalizations with mathematical precision. By being an expressive alternative foundation for mathematics it enables the formalization of constructive proofs through the connections to intuitionistic logic given by the Brouwer–Heyting–Kolmogorov interpretation. After creating a formalization by defining its types and their elements, one can express statements and theorems in forms of new types, the instances of which can be thought of as proofs for them. This is due to the so called “*propositions-as-types*” paradigm, also known as the Curry–Howard isomorphism. Its use cases include examples such as constructing algebraic descriptions of programming languages [4] and proving the correctness of programs implemented in them. Some of these procedures are already possible in currently available proof assistant systems (such as Agda [7][1]), but certain steps of them are rather cumbersome. They either require plenty of manual work (due to the need of eliminating equalities using transports) when utilizing certain constructions, such as quotient types, or on the other hand, if such features (e.g. rewrite rules) are utilized that prevent the code from getting overly complicated, important properties of type theory - such as canonicity and normalizations - are lost.

This issue could be mitigated by creating a proof assistant based on Setoid Type Theory [3][2], which has certain features that are missing from Martin-Löf Type Theory, such as quotient types, functional extensionality and propositional extensionality, which makes handling the previously mentioned problems much more convenient and elegant. As the usual naming convention suggests, the interpretations of contexts and closed types in the setoid model are setoids, which are formed by extending a set with a reflexive, symmetric and transitive equivalence relation.

2 Requirements

In order to utilize Setoid Type Theory for such purposes, it is required that we first show that it satisfies certain basic requirements, such as canonicity and decidable equality, which is necessary for type checking. The proof that these conditions are met can be constructed in several different ways, among which I have examined some methods, comparing their advantages and disadvantages.

3 Methods

Setoid Type Theory itself can be defined through multiple approaches, in the current literature [3][2] there is no consistent description yet, which publications agree on. We are looking for rules that are (in certain aspects) optimal for defining the basic framework, on top of which proofs and implementations can be built. Such beneficial rules can be indicated by and extracted from models, in which desired properties are convenient to prove.

Canonicity can be proven directly using the method of logical predicates [5], which is a more involved procedure, or indirectly through defining a setoid model based on another model of type theory. The latter procedure is called *setoidification*, which is illustrated on figure 1. In that case we can derive the properties of the syntax of Setoid Type Theory from the properties of the original model if the mapping to the newly created model is shown to be injective.

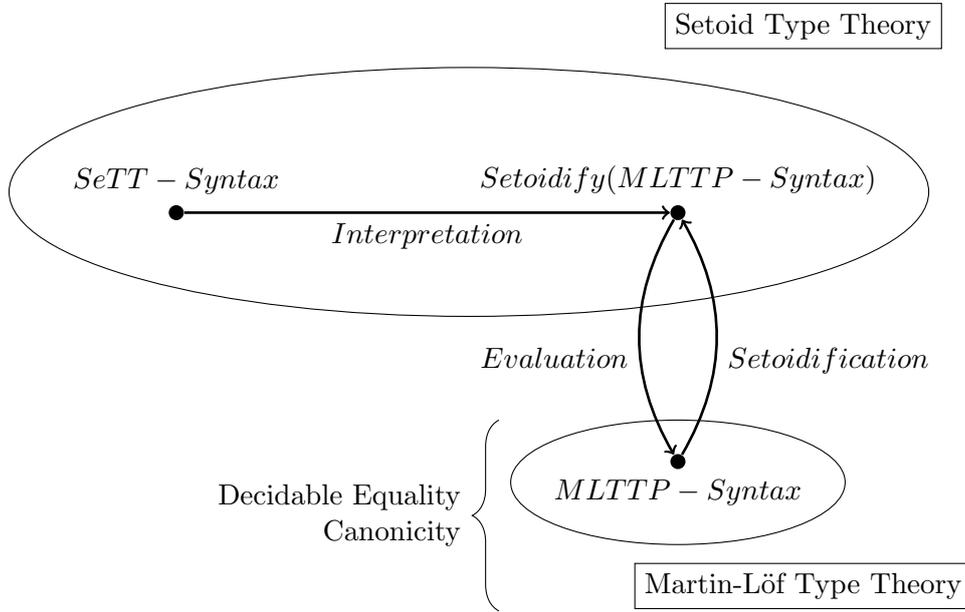


Figure 1: Setoidification illustrated

Let the $\llbracket _ \rrbracket$ operation be the interpretation from the syntax of SeTT to the setoidified MLTTP syntax. If the injectivity of this bracket function ($\llbracket t \rrbracket = \llbracket t' \rrbracket \Rightarrow t = t'$) is proven, the desired properties can be lifted to SeTT by evaluating to the underlying model.

- *Decidability of equality*

Decidability of equality means that either $t = t'$ or $t \neq t'$ holds for every t, t' . Through evaluation, $\llbracket t \rrbracket = \llbracket t' \rrbracket$ is an equality in the base model. If we have decidable equality there, we have two cases:

- $\llbracket t \rrbracket = \llbracket t' \rrbracket$ implies $t = t'$ by injectivity
- $\llbracket t \rrbracket \neq \llbracket t' \rrbracket$ implies $t \neq t'$ by contradiction

- *Canonicity*

Canonicity means that every closed term can be equated to one of potentially several specific forms defined by its type, which are usually considered as final results of computations. For example, type `Bool` has two canonical forms in the base model:

- $\llbracket t \rrbracket = \text{MLTTP} - \text{Syntax.false}$ implies $t = \text{SeTT} - \text{Syntax.false}$ by injectivity
- $\llbracket t \rrbracket = \text{MLTTP} - \text{Syntax.true}$ implies $t = \text{SeTT} - \text{Syntax.true}$ by injectivity

The proof of this injectivity can also be simplified if the definition of the setoidification conforms to certain constraints given by the notion of *"interpretation"* [6], more or less meaning that the carrier of the setoids stays the same as the set in the base model, upon which the construction is built.

4 Contribution

We described several different models with homogeneous and heterogeneous equalities, symmetry and transitivity, bidirectional coercion as well as other differing aspects and observed their properties in order to determine a convenient set of rules to be used. Some of the models were formally implemented in the Agda proof assistant as well and typechecked by computer.

We concluded that a model with bidirectional coercion instead of symmetry and transitivity is more generic, but requires a higher dimensional reasoning, which makes it less straightforward to work with. Homogeneous equality on the other hand would reduce the need to express equalities of contexts and simplify the model, but in order to keep the same level of expressive power, new rules (at least weak fibrancy) need to be introduced, while the asymmetrical nature of coercion - that arises at for example the definition of the sigma type - led to extra complications.

One other choice we made was having a separate family of types and terms for propositions instead of encoding them into universe levels, since this is a more generic approach, as this way it can exist in the presence of universes and their absence as well.

We created two setoidification constructions, for one of them the injectivity proof is already in progress.

5 Further work

Based on our results and experiences so far the model with the following properties seems to have the best in terms of usability and prospective simplicity of proofs:

- Heterogeneous equality
- Symmetry and transitivity
- Separate family of propositional types and terms

We plan to compose a canonicity proof for a setoidification model through the injectivity of the construction for such a minimal Setoid Type Theory extended with a few basic type formers, such as Σ , Π and `Bool` with large eliminator.

References

- [1] **Agda Development Team** Agda: A dependently typed functional programming language and proof assistant <https://github.com/agda/agda>
- [2] **Altenkirch, Thorsten and Boulier, Simon and Kaposi, Ambrus and Sattler, Christian and Sestini, Filippo** Constructing a universe for the setoid model in: *Foundations of Software Science and Computation Structures - 24th International Conference, FOSSACS 2021, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2021, Luxembourg City, Luxembourg, March 27 - April 1, 2021*
- [3] **Altenkirch, Thorsten and Boulier, Simon and Kaposi, Ambrus and Tabareau, Nicolas** Setoid Type Theory—A Syntactic Translation in: *Mathematics of Program Construction*
- [4] **Altenkirch, Thorsten and Kaposi, Ambrus** Type Theory in Type Theory Using Quotient Inductive Types in: Proceedings of the 43rd Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages St. Petersburg, FL, USA
- [5] **Kaposi, Ambrus and Huber, Simon and Sattler, Christian** Gluing for Type Theory in: 4th International Conference on Formal Structures for Computation and Deduction (FSCD 2019)
- [6] **Moeneclaey, Hugo** Parametricity and Semi-Cubical Types, 2021 in: arXiv, Computing Research Repository <https://arxiv.org/abs/2105.08422>
- [7] **Norell, Ulf** Dependently typed programming in Agda, Springer, *International school on advanced functional programming* 2008

A model of type theory supporting quotient inductive-inductive types¹

Ambrus Kaposi and Zongpu Xie

Eötvös Loránd University

akaposi@inf.elte.hu and szumixie@gmail.com

Quotient inductive-inductive types

Natural numbers, lists, binary trees, the syntax of terms of a programming language are all examples of inductive types. In general, an inductive type is a collection of trees of finite depth with a fixed branching structure. Inductive-inductive types (IITs) [11] generalise inductive types by allowing multiple mutually defined sorts where the later sorts can be indexed over the previous ones. Quotient inductive-inductive types (QIITs) [9] in addition allow equality constructors. They are more general than simple quotients as the elements of the type are generated *at the same time* as the quotienting. This allows the constructive definition of Cauchy real numbers in type theory without using the axiom of choice [12]. Other examples are the partiality monad [3], the intrinsic syntax of programming languages [4]. In general, for any generalised algebraic theory [5], the initial object in the category of its models is a QIIT [9].

A simple example of a QIIT is a subset of the intrinsic syntax of type theory [4] with two sorts, five operators and one equation:

$$\begin{aligned}
 \text{Con} & : \text{Set} \\
 \text{Ty} & : \text{Con} \rightarrow \text{Set} \\
 \bullet & : \text{Con} \\
 - \triangleright - & : (\gamma : \text{Con}) \rightarrow \text{Ty } \gamma \rightarrow \text{Con} \\
 \text{U} & : \text{Ty } \gamma \\
 \text{El} & : \text{Ty } (\gamma \triangleright \text{U}) \\
 \Sigma & : (a : \text{Ty } \gamma) \rightarrow \text{Ty } (\gamma \triangleright a) \rightarrow \text{Ty } \gamma \\
 \text{eq} & : \gamma \triangleright \Sigma a b = \gamma \triangleright a \triangleright b
 \end{aligned}$$

There is an ordinary sort of contexts Con , a sort of types Ty indexed over contexts. That is, for every $\gamma : \text{Con}$ we have that $\text{Ty } \gamma$ is a sort. There are two operators producing contexts, empty context \bullet and context extension $- \triangleright -$, the latter refers to types. This is why Ty cannot be defined separately after Con , they have to be given mutually, at the same time. There are three different ways to form types corresponding to the three operators

¹The first author was by supported by the “Application Domain Specific Highly Reliable IT Solutions” project which has been implemented with the support provided from the National Research, Development and Innovation Fund of Hungary, financed under the Thematic Excellence Programme TKP2020-NKA-06 (National Challenges Subprogramme) funding scheme. The second author was supported by the European Union, co-financed by the European Social Fund (EFOP-3.6.3-VEKOP-16-2017-00002).

producing Tys. Finally, the equation `eq` says that extending a context γ with a Σ type made of a and b is the same as extending it first with a and then with b .

A model of type theory supports a particular QIIT if (i) there is an algebra called the constructors: this means that for each sort there is a type, for each operator there are corresponding terms and for each equation there is a term of the corresponding identity type; (ii) for any other algebra there is a unique algebra homomorphism from the constructor to this algebra. An algebra homomorphism is given by terms (functions) for each sort that respect the operations up to definitional equality of the model. If they only respect the operations up to the identity type of the model, we only have propositional (weak) computation rules for the QIIT. A formal description of these notions is given in [9].

The universal QIIT

If a model of extensional type theory (ETT) supports a particular QIIT called *the universal QIIT* (UQIIT), then it supports all QIITs [9]. Hofmann’s conservativity theorem [8] says that if a type can be expressed in intensional type theory with function extensionality and uniqueness of identity proofs (ITT+funext+UIP) and there is a term of this type in ETT, then there is also a term of this type in ITT+funext+UIP. As for any QIIT Ω the type

“the universal QIIT exists \Rightarrow Ω exists”

can be expressed in ITT+funext+UIP, we can transfer the proof of [9] given in ETT to any model of ITT+funext+UIP. However as this type expresses the computation rules in “ Ω exists” using the identity type of the model instead of definitional equality, we only obtain weak computation rules. Thus we have that if a model of ITT+funext+UIP supports the UQIIT with propositional computation rules, then it supports all QIITs with propositional computation rules.

The setoid model

The setoid model of type theory [1] justifies funext and UIP and our goal is to show that it also supports the UQIIT, thus all QIITs with propositional computation rules.

We formalised in Agda the setoid model as a category with families (CwF [6]) with extra structure. We formalised what it means that this model supports the UQIIT. More precisely, we formalised what it means for a strict model to support the UQIIT. We typechecked our formalisation with two strict models: the setoid model and the standard (set) model. This involves saying what types and terms a UQIIT algebra consists of, what a UQIIT homomorphism between two such algebras is (with definitional computation rules) and what it means that two parallel homomorphisms are equal (up to the model’s identity type). We used a variant of the UQIIT which supports infinitary operations [10]. The formalisation is available at the URL <https://bitbucket.org/akaposi/qiit>.

In the setoid model a context is given by a set (a type in the metatheory or target model) and an equivalence relation. A type is a family of sets together with a family of equivalence relations indexed over those of the context and fibration conditions: if there are two elements of the context that are related, then we can transport between types at each element (coercion), and this transport preserves the relation of the type (coherence). A term is a function between the sets which respects the relation. In the setoid model,

the identity type is not given by an inductive type once and for all for all types, but by a separate relation for each type. This relation is inductively generated for inductive types and is coinductively generated for coinductive types. Specifically, the identity type of two functions (the function type is a coinductive type) says that they are pointwise equal (pointwise equality is another function, hence a coinductive type). From the reflexive property of the equivalence relation we obtain the usual reflexivity constructor of the identity type. From the fact that all terms respect equality and the coercion operation of types, we obtain the usual eliminator (transport or J) of the identity type. From the definition of the relation for function space, we obtain function extensionality and from the fact that the equivalence relations are proof irrelevant (target **Prop** [7] instead of **Set**), we obtain uniqueness of identity proofs. Thus setoids model ITT+funext+UIP. Moreover, the setoid model is strict, that is, all equalities are definitional in type theory as metatheory. This provides a *model construction* (syntactic translation) [2]: from any model of ITT with **Prop**, we obtain a “setoidified” version of the model which satisfies funext and UIP.

QIITs in the setoid model

In the setoid model, we can define the constructors of the UQIIT. This is given by an IIT with twice as many sorts as the UQIIT: there is a “point sort” for each sort of the QIIT and an additional propositional sort for its equality. The point sorts have a constructor for each operation and the indexed point sorts in addition have coercion constructors. The equality sorts are the free fibrant equivalence congruence relations over the equality constructors: all of them have constructors for reflexivity, symmetry, transitivity, one congruence constructor for each operator, one constructor for each equation, and for the indexed sorts, there is a coherence constructor. For example, we can define our example QIIT with **Con** and **Ty** in the setoid model using the following IIT (sometimes we denote implicit arguments by curly braces, e.g. in the type of \sim_U).

$ \mathbf{Con} $	$: \mathbf{Set}$	point sorts
$ \mathbf{T}_y $	$: \mathbf{Con} \rightarrow \mathbf{Set}$	\vdots
$-\sim_{\mathbf{Con}}-$	$: \mathbf{Con} \rightarrow \mathbf{Con} \rightarrow \mathbf{SProp}$	equality sorts
$\sim_{\mathbf{T}_y}$	$: \gamma \sim_{\mathbf{Con}} \gamma' \rightarrow \mathbf{T}_y \gamma \rightarrow \mathbf{T}_y \gamma' \rightarrow \mathbf{SProp}$	\vdots
$ \bullet $	$: \mathbf{Con} $	point constructor for
$-\triangleright-$	$: (\gamma : \mathbf{Con}) \rightarrow \mathbf{T}_y \gamma \rightarrow \mathbf{Con} $	each operator
$ \mathbf{U} $	$: \mathbf{T}_y \gamma$	\vdots
$ \mathbf{E} $	$: \mathbf{T}_y (\gamma \triangleright \mathbf{U})$	
$ \Sigma $	$: (a : \mathbf{T}_y \gamma) \rightarrow \mathbf{T}_y (\gamma \triangleright a) \rightarrow \mathbf{T}_y \gamma$	
\sim_\bullet	$: \bullet \sim_{\mathbf{Con}} \bullet $	congruence for
$-\sim_{\triangleright}-$	$: (\bar{\gamma} : \gamma \sim_{\mathbf{Con}} \gamma') \rightarrow \sim_{\mathbf{T}_y} \bar{\gamma} \alpha \alpha' \rightarrow (\gamma \triangleright a) \sim_{\mathbf{Con}} (\gamma' \triangleright a')$	each operator
\sim_U	$: \{\bar{\gamma} : \gamma \sim_{\mathbf{Con}} \gamma'\} \rightarrow \sim_{\mathbf{T}_y} \bar{\gamma} (\mathbf{U} \{\gamma\}) (\mathbf{U} \{\gamma'\})$	\vdots
$\sim_{\mathbf{E} }$	$: \{\bar{\gamma} : \gamma \sim_{\mathbf{Con}} \gamma'\} \rightarrow \sim_{\mathbf{T}_y} \bar{\gamma} (\mathbf{E} \{\gamma\}) (\mathbf{E} \{\gamma'\})$	
\sim_Σ	$: (\bar{a} : \sim_{\mathbf{T}_y} \bar{\gamma} a a') \rightarrow \sim_{\mathbf{T}_y} (\bar{\gamma} \sim_{\triangleright} \bar{a}) b b' \rightarrow$ $\sim_{\mathbf{T}_y} \bar{\gamma} (\Sigma a b) (\Sigma a' b')$	

$ \text{eq} $	$:\ \gamma \triangleright \Sigma a\ b \sim_{\text{Con}} \gamma \triangleright a \triangleright b$	equality constructor
refl_{Con}	$:\ (\gamma : \text{Con}) \rightarrow \gamma \sim_{\text{Con}} \gamma$	equivalence relations
sym_{Con}	$:\ \gamma \sim_{\text{Con}} \gamma' \rightarrow \gamma' \sim_{\text{Con}} \gamma$	\vdots
$\text{trans}_{\text{Con}}$	$:\ \gamma \sim_{\text{Con}} \gamma' \rightarrow \gamma' \sim_{\text{Con}} \gamma'' \rightarrow \gamma \sim_{\text{Con}} \gamma''$	
refl_{Ty}	$:\ (a : \text{Ty} \gamma) \rightarrow \sim_{\text{Ty}} (\text{refl}_{\text{Con}} \gamma) a a$	
sym_{Ty}	$:\ \sim_{\text{Ty}} \bar{\gamma} a a' \rightarrow \sim_{\text{Ty}} (\text{sym}_{\text{Con}} \bar{\gamma}) a' a$	
trans_{Ty}	$:\ \sim_{\text{Ty}} \bar{\gamma} a a' \rightarrow \sim_{\text{Ty}} \bar{\gamma}' a' a'' \rightarrow \sim_{\text{Ty}} (\text{trans}_{\text{Con}} \bar{\gamma} \bar{\gamma}') a a''$	
coe_{Ty}	$:\ \gamma \sim_{\text{Con}} \gamma' \rightarrow \text{Ty} \gamma \rightarrow \text{Ty} \gamma'$	fibration conditions
coh_{Ty}	$:\ (\bar{\gamma} : \gamma \sim_{\text{Con}} \gamma') (\alpha : \text{Ty} \gamma) \rightarrow \sim_{\text{Ty}} \bar{\gamma} \alpha (\text{coe}_{\text{Ty}} \bar{\gamma} \alpha)$	\vdots

We define the IIT for the UQIIT completely analogously. Then the constructor UQIIT algebra is given exactly by the components of this IIT.

The recursor takes as input a UQIIT algebra in the empty context and returns a homomorphism from the constructors to this algebra. The computation rules are definitional. An algebra in an arbitrary context can be turned into an algebra in the empty context \cdot . For example, for a type $\Gamma \vdash C$, we turn it into $\cdot \vdash \Pi(x : \mathbf{K} \Gamma).C[x]$, that is to a dependent function type where the domain is constant (\mathbf{K}) Γ . Thus, when we want to use the recursor in a context Γ , we convert the algebra into an algebra in the empty context, and then we apply the recursor. The computation rules of this lifted recursor are still definitional. Uniqueness of the recursor and the fact that the recursor is stable under substitution are proved by induction on the IIT.

Further work

The notion of UQIIT algebra, morphism, the implementation IIT and the recursor in the empty context are all defined in Agda. The lifting of the recursor to arbitrary context is still a work in progress.

An alternative of our construction would be to directly reduce a QIIT to an IIT in the setoid model by induction on the QIIT signature. This way we would avoid going through the UQIIT and ETT and we expect that we would get definitional computation rules.

References

- [1] Thorsten Altenkirch. Extensional equality in intensional type theory. In *14th Annual IEEE Symposium on Logic in Computer Science, Trento, Italy, July 2-5, 1999*, pages 412–420. IEEE Computer Society, 1999.
- [2] Thorsten Altenkirch, Simon Boulier, Ambrus Kaposi, and Nicolas Tabareau. Setoid type theory—a syntactic translation. In Graham Hutton, editor, *Mathematics of Program Construction*, pages 155–196, Cham, 2019. Springer International Publishing.
- [3] Thorsten Altenkirch, Nils Anders Danielsson, and Nicolai Kraus. Partiality, revisited. In *Proceedings of the 20th International Conference on Foundations of Software Science and Computation Structures - Volume 10203*, page 534–549, Berlin, Heidelberg, 2017. Springer-Verlag.

- [4] Thorsten Altenkirch and Ambrus Kaposi. Type theory in type theory using quotient inductive types. In Rastislav Bodik and Rupak Majumdar, editors, *Proceedings of the 43rd Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, POPL 2016, St. Petersburg, FL, USA, January 20 - 22, 2016*, pages 18–29. ACM, 2016.
- [5] John Cartmell. Generalised algebraic theories and contextual categories. *Annals of Pure and Applied Logic*, 32:209–243, 1986.
- [6] Peter Dybjer. Internal type theory. In *Lecture Notes in Computer Science*, pages 120–134. Springer, 1996.
- [7] Gaëtan Gilbert, Jesper Cockx, Matthieu Sozeau, and Nicolas Tabareau. Definitional proof-irrelevance without K. *Proc. ACM Program. Lang.*, 3(POPL):3:1–3:28, 2019.
- [8] Martin Hofmann. Conservativity of equality reflection over intensional type theory. In *TYPES 95*, pages 153–164, 1995.
- [9] Ambrus Kaposi, András Kovács, and Thorsten Altenkirch. Constructing quotient inductive-inductive types. *Proc. ACM Program. Lang.*, 3(POPL):2:1–2:24, January 2019.
- [10] András Kovács and Ambrus Kaposi. Large and infinitary quotient inductive-inductive types. In Holger Hermanns, Lijun Zhang, Naoki Kobayashi, and Dale Miller, editors, *LICS '20: 35th Annual ACM/IEEE Symposium on Logic in Computer Science, Saarbrücken, Germany, July 8-11, 2020*, pages 648–661. ACM, 2020.
- [11] Fredrik Nordvall Forsberg. *Inductive-inductive definitions*. PhD thesis, Swansea University, 2013.
- [12] The Univalent Foundations Program. Homotopy type theory: Univalent foundations of mathematics. Technical report, Institute for Advanced Study, 2013.

Staged Compilation and Generativity

András Kovács

Eötvös Loránd University, Department of Programming Languages and Compilers
kovacsandras@inf.elte.hu

Overview

The purpose of staged compilation is to write code-generating programs in a safe and ergonomic way. Although it is always possible to write metaprograms by simply manipulating strings or deeply embedded syntax trees, this is often error-prone and tedious. Staging is a way to have more guarantees about the safety and well-typing of metaprograms, and also a way to integrate object-level and meta-level syntaxes more organically.

Two-level type theory (2LTT) [2] was originally developed for the purpose of doing synthetic homotopy theory, by adding a metaprogramming layer on top of homotopy type theory [6]. However, it turns out that 2LTT is also a great framework for metaprogramming and staging in general, and it is applicable to a wide range of theories, both on the object and meta level.

There is a simple semantics to 2LTT which justifies the metaprogramming view: this is the *presheaf model* of 2LTT. Here, meta-level types are presheaves over the underlying category of the object theory. Hence, every meta-level construction must be stable under the object-level morphisms. The advantage of working in 2LTT stems from stability under morphisms: if every construction is automatically stable, it becomes possible to omit explicit handling of base morphisms. More concretely, from the staging perspective, this means that we never have to deal with scoping, renaming, substitution or de Bruijn indices in the object syntax, when working in 2LTT.

Generativity. Generativity means that we can only generate code, but not make decisions based on the internal structure of object-level syntax. Generativity simplifies staging, and it is often enforced in practical implementations [5]. However, non-generative staging provides additional power and flexibility. A simple example for a non-generative feature is conversion checking. This can be viewed as an axiom in 2LTT, which says that meta-level (“strict”) equality of object-level values is decidable. We aim to investigate semantics of non-generativity in the following.

Basic Rules and Usage of 2LTT

To illustrate using 2LTT for staging, we specify a simple variant of 2LTT where we have exactly the same dependent type theory for the object-level and meta-level theories.

We have universes U_i^s , where $s \in \{0, 1\}$, denoting a *stage* or level in the 2LTT sense, and $i \in \mathbb{N}$ denotes a usual level index of *sizing* hierarchies. The two dimensions of indexing are orthogonal, and we will elide the i indices in the following. We assume Russell-style universes. Both U_0 and U_1 may be closed under arbitrary type formers, but eliminators

in each *only target the same universe*, i.e. elimination cannot cross universes. We have the following operations:

- For $A : U_0$, we have $\text{Code } A : U_1$. This is the type of meta-level programs which return object-level code with type A .
- Quoting: for $A : U_0$ and $t : A$ we have $\langle t \rangle : \text{Code } A$. In other words, for every object-level term we have a metaprogram which immediately returns that term.
- Splicing: for $t : \text{Code } A$, we have $\sim t : A$. This means running a metaprogram and inserting its result into object-level code.
- We also know that quoting/splicing is a definitional isomorphism, so $\langle \sim t \rangle = t$ and $\sim \langle t \rangle = t$.

A *staging algorithm* takes as input a closed term $t : A$ where $A : U_0$, and splices the results of all metaprograms, so that we get an output term which is free of splices. This can be implemented using variations of normalization-by-evaluation [1] which track current stages. We do not detail staging algorithms here.

Let's look at some examples. We have the object-level identity function as usual:

$$\begin{aligned} \text{id}_0 &: (A : U_0) \rightarrow A \rightarrow A \\ \text{id}_0 &:= \lambda A x. x \end{aligned}$$

Staging does not do anything with id_0 , since it has no splices. Likewise if we apply id_0 to object-level values, as in $\text{id}_0 \text{Bool true}$. We also have the meta-level version:

$$\begin{aligned} \text{id}_1 &: (A : U_1) \rightarrow A \rightarrow A \\ \text{id}_1 &:= \lambda A x. x \end{aligned}$$

Note that this also works on object-level values, because of quoting:

$$\sim(\text{id}_0 (\text{Code Bool}_0) \langle \text{true}_0 \rangle) : \text{Bool}_0$$

Staging the above term computes to $\sim \langle \text{true}_0 \rangle$, which in turn computes to true_0 . Thus, id_1 is compile-time evaluated. There's a third version, which is a specialized version of id_1 : it's also evaluated at compile time, but it only works on object-level types:

$$\begin{aligned} \text{id}_{\text{Code}} &: (A : \text{Code } U_0) \rightarrow \text{Code } (\sim A) \rightarrow \text{Code } (\sim A) \\ \text{id}_{\text{Code}} &:= \lambda A x. x \end{aligned}$$

Now, $\sim(\text{id}_{\text{Code}} (\text{Code Bool}_0) \langle \text{true}_0 \rangle)$ also stages to true_0 . Meta-functions which are restricted to Code are also useful when we want to define functions which are partially evaluated at compile time. For example, if we want to inline a function argument for object-level list mapping:

$$\begin{aligned} \text{map} &: (A B : \text{Code } U_0) \rightarrow (\text{Code } (\sim A) \rightarrow \text{Code } (\sim B)) \\ &\rightarrow \text{Code}(\text{List}_0 (\sim A)) \rightarrow \text{Code}(\text{List}_0 (\sim B)) \\ \text{map} &:= \lambda A B f as. \langle \text{foldr}_0 (\lambda a bs. \text{cons}_0 (\sim (f \langle a \rangle)) bs) \text{nil}_0 (\sim as) \rangle \end{aligned}$$

Presheaf Model

Why consider the presheaf model? The reason is that it's the simplest semantics which justifies the metaprogramming view of 2LTT. It is *not* the same thing as the staging algorithm, which is based on normalization-by-evaluation, which is much more complicated to formalize [3]. The presheaf model is fairly simple as far as models go, so it's interesting to see which staging features can be justified with it. We skip presenting the whole presheaf model. For details, we refer the reader to [4, Section 1.2].

We give some examples for interpreting constructions in the model. We present the results up to isomorphism, with some simplifications. We assume now that object-level morphisms are substitutions. We have $\mathbf{Bool}_0 : \mathbf{U}_0$ and $\mathbf{Bool}_1 : \mathbf{U}_1$.

- A closed function $t : \mathbf{Bool}_1 \rightarrow \mathbf{Bool}_1$ becomes a metatheoretical function in $\mathbb{B} \rightarrow \mathbb{B}$.
- A closed function $t : \mathbf{Bool}_0 \rightarrow \mathbf{Bool}_0$ becomes a closed object-theoretic function in $\mathbf{Bool} \rightarrow \mathbf{Bool}$.
- A closed function $t : \mathbf{Code}\mathbf{Bool}_0 \rightarrow \mathbf{Bool}_1$ becomes a function which maps a \mathbf{Bool} term in any context to \mathbb{B} , such that the function commutes with object-theoretic substitution. For example if we have a variable x , we can substitute it with any term before feeding it to the semantic t , and the result is the same. In fact, this means that t cannot depend on its term argument, hence t is specified simply by a \mathbb{B} .
- A closed function $t : \mathbf{Bool}_1 \rightarrow \mathbf{Code}\mathbf{Bool}_0$ becomes simply a pair of closed \mathbf{Bool} terms.

Yoneda lemma. The Yoneda lemma is a general statement which restricts the way meta-level values can depend on object-level ones. First, note that any object-level typing context Γ can be mapped to a presheaf, by taking the sets of parallel substitutions into Γ . This is the *Yoneda embedding* of Γ , denoted by $y\Gamma$. The Yoneda lemma says that we have the following isomorphism of sets:

$$(y\Gamma \Rightarrow \Delta) \simeq |\Delta|\Gamma$$

where \Rightarrow means a natural transformation, and $|\Delta|\Gamma$ denotes the set that we get by evaluating the Δ presheaf at the object-level Γ context. From this, what we essentially get is that any 2LTT term $\Gamma \vdash t : A$, such that Γ is essentially interpreted as $y\Gamma'$ for some Γ' , is interpreted as an element of $|A|\Gamma'$. We call Γ *representable* if there is such Γ' .

In particular, if $\Gamma \vdash t : \mathbf{Bool}_1$ and Γ representable, then since $|\mathbf{Bool}_1|\Gamma' = \mathbb{B}$, t is simply an element of \mathbb{B} in the semantics, and cannot depend on the typing context.

In short, whether the Yoneda lemma applies to a given term, depends on whether the typing context is representable. In turn, the representability of the context depends on what morphisms are in the object theory. We consider two options.

1. Morphisms are substitutions. In this case, y preserves context extension, i.e. $y(\Gamma, x : A) \simeq (y\Gamma, x : yA)$ in the presheaf model. That's because a substitution which targets $(\Gamma, x : A)$ is equivalent to a pair of substitutions, targeting Γ and A respectively. Therefore, if we have $x_1 : A_1, x_2 : A_2, \dots, x_i : A_i \vdash t : B$, such that all A_i are representable, the entire context is also representable, and the Yoneda lemma applies.

Which types are representable? For starters, every type of the form `Code A`, since `Code` in the model is essentially interpreted as γ (eliding the formal complications arising from possible dependencies of A on the context). Also, if we have $x : A$ in a context, where $A : \mathcal{U}_0$, that context extension is also interpreted as extension with γA . In short: if the context only has `Code` types or types in \mathcal{U}_0 , it is representable.

This greatly limits non-generative features in the model. Consider adding the axiom which says that meta-level equality of object-level values is decidable: $x : \text{Code } A, y : \text{Code } A \vdash \text{conversion}_A : (x = y) + (x = y \rightarrow \perp)$. This is a simple non-generative axiom, since any constructive interpretation must look inside `Code`-s. This axiom is false if object morphisms are substitutions. That’s because the context is representable, so we can simplify using the Yoneda lemma. The statement that we get in the model is that “two terms are either equal, or they are unequal and remain unequal after arbitrarily substitutions”. Now, if we pick two *variables* x and y such that $x \neq y$, then they are not equal, but they can be also made equal by substituting both variables with the same term.

Can we repair this? One possibility is to have decidable equality only for *closed* terms. However, the syntax of 2LTT provides no way to talk about closed terms. Instead we’d have to use a closed modality [3]. This would be interesting to investigate in future work.

2. Morphisms are weakenings. In this case, object morphisms are so-called *order-preserving embeddings*, meaning that a morphism can drop zero or more entries from a context, so morphisms are essentially bitmasks which mark a sub-context. The action of weakening embeds terms in larger contexts. Moreover, γ does not preserve context extension. Hence, typing contexts are not necessarily representable, even if they only contain object-level bindings. So the Yoneda-reduction of dependencies generally does not apply.

Now, `conversion` is fine, because inequality of terms is stable under weakening. On the other hand, by only having weakening in the object theory, the range of supported object theories is greatly restricted. For example, we can’t have β -reduction for functions in the equational theory, since that’s specified using substitution. Likewise, dependent types are out, since the typing of dependent elimination involves substitution.

Simple type theories still work, if we only have weakening in their equational theory. For the perspective of staging, this is fine, because in *code generation* we care about the intensional definition of programs, and we do not want to equate β -reducts, since a primary use-case of staging is to improve runtime performance, hence distinguish between possibly β -convertible programs.

It appears that if morphisms are weakenings, then the presheaf model is compatible with a wide range of non-generative axioms. For example, we can also postulate *countability* of `Code A`, i.e. that there are injections $\text{index}_A : \text{Code } A \rightarrow \text{Nat}_1$. In the presheaf model, the indexing function works by enumerating maximally strengthened terms, which are stable under weakening.

Other ways of justifying non-generativity. An alternative solution would be to use something other than the presheaf model to justify non-generative axioms. For example: could we use the staging algorithm itself, i.e. does normalization-by-evaluation support non-generativity? It seems likely, as semantic values need only be stable under weakening. This remains future work.

References

- [1] Andreas Abel. *Normalization by Evaluation: Dependent Types and Impredicativity*. PhD thesis, Ludwig-Maximilians-Universität München, 2013. Habilitation thesis.
- [2] Danil Annenkov, Paolo Capriotti, Nicolai Kraus, and Christian Sattler. Two-level type theory and applications. *ArXiv e-prints*, may 2019.
- [3] Rafaël Bocquet, Ambrus Kaposi, and Christian Sattler. Induction principles for type theories, internally to presheaf categories. *arXiv preprint arXiv:2102.11649*, 2021.
- [4] Simon Huber. *Cubical Interpretations of Type Theory*. PhD thesis, University of Gothenburg, 2016.
- [5] Oleg Kiselyov. The design and implementation of BER metaocaml - system description. In Michael Codish and Eijiro Sumii, editors, *Functional and Logic Programming - 12th International Symposium, FLOPS 2014, Kanazawa, Japan, June 4-6, 2014. Proceedings*, volume 8475 of *Lecture Notes in Computer Science*, pages 86–102. Springer, 2014.
- [6] The Univalent Foundations Program. *Homotopy Type Theory: Univalent Foundations of Mathematics*. <https://homotopytypetheory.org/book>, Institute for Advanced Study, 2013.



Section:

Using artificial intelligence tools in molecular structure prediction:

The Budapest Amyloid Predictor and its applications

Organizer: Vince Grolmusz

Invited talk:

András Perczel: The amyloid state of proteins

Contributions:

- László Keresztes: Amyloid patterns in hexapeptides
- Evelin Szögi: Predicting the amyloid state by Support Vector Machines
- Kristóf Takács and Vince Grolmusz: Sliding windows in the Protein Data Bank: amyloid-forming propensity of prefixes and suffixes of secondary structures
- Bálint Varga: Pathfinding in the hexapeptide-graph: through the amyloid and non-amyloid nodes

The amyloid state of proteins

András Perczel

MTA-ELTE Protein Modeling Research Group & Laboratory of Structural Chemistry
and Biology, Institute of Chemistry, ELTE, Budapest, Hungary

`perczel.andras@ttk.elte.hu`

In developed countries age related diseases are in focus, as life expectancy increases and everyone wishes a better quality of life. Aging-related diseases are unquestionably connected to protein aggregation, more precisely to amyloid formation as is the case of cataracts, type II Diabetes Mellitus, Alzheimer's disease etc.. Formation of protein fibrils, plaques and tangles are typical markers of the above and other diseases highlighting that behind the neuroanatomic changes established postmortem: key proteins undergo restructuring. Today over 50 different polypeptides and proteins are known to self-assemble and form fatal amyloid fibrils. Out of the approximately 100 000 proteins of an eukaryotic cell 70% is built up from "independent" domains or modules. When these linear polymer are module-like, they autonomously fold into a globular structure, whose conformation correlates strongly with its biological function. The relative abundance of open conformer(s) can be enhanced by mutation(s), which in turn can lead to amyloid formation, aggregation and deposition of the protein. For example, in lysozyme a point mutation can initiate such a deposition. (Booth et al., Nature, 1997, 385, 787) Deciphering intermediates and "open" forms of proteins by NMR (Rovó et al. Chemistry a European Journal 2013, 19, 2628) is therefore a key approach to understand the rout to amyloid formation. Hundreds of experimental evidences show that beside the unique globular fold, proteins can thus have an alternative, highly structured, fibrillar form (Dobson Trends Biochem. Sci. 1999). Amyloid like fibrils are built up from repeated β -sheets, which are in turn made up of β -strands orthogonal to their main axis. (Sunde & Blake, Adv. Prot. Chem. 1997) The formation of these "amyloid" like β -strands is not coded by any specific amino acid sequence! Therefore the question to be asked seems obvious, namely is aggregation a natural or an abnormal process? Is it possible that the "amyloid" like aggregates of proteins are generic and intrinsic structures of polypeptides composed of natural alpha amino acids? If the formation of an amyloid like fibril is indeed not coded by the amino acid sequence, then its formation cannot be governed by the amino acid side-chains. Therefore, the aggregation is due to the interaction between backbone atoms! The stability of the supramolecular complex increases both with the length of the polypeptide chain and the number of interacting β -strands increasing (Beke et al. JACS 2006, Pohl et al. JACS 2006, Perczel et al. JACS 2007, Horváth et al. 2019) The formation of an "amyloid" like supramolecular "matrix" from β -sheets is energetically favored. Thus, the aggregation of polypeptides is indeed a "normal" energy driven process. Proteins can be regarded as "misfolded" polymers, or perhaps proteins are "misfolded" amyloids?! Conclusion: 1) The formation of amyloid like β -strands is not coded by any specific amino acid sequence. 2) The H-bond β -layers are characteristics of amyloids. 3) Both length and the number of amino acids in the extended polypeptide chain makes β -layer more stable. 4) Amyloid type aggregation of polypeptides is indeed a normal, energy driven process. Amino acid composition and the form of the dry steric zipper as interfaces fine tunes this natural process.



Amyloid patterns in hexapeptides

László Keresztes

Institute of Mathematics, Eötvös Loránd University

laszlo.keresztes@student.elte.hu

Motivation

We achieved great classification accuracy with SVM in the amyloid classification task of hexapeptides.[1] We were interested in how could we extract information about what drives the amyloid property in hexapeptides.

Amyloid Effect Matrix

Using a linear classifier (e.g. SVM), one can construct a matrix of size 20×6 , which compresses the classification method.

	1	2	3	4	5	6
A	-0.26	-0.32	-0.27	-0.14	-0.43	-0.22
R	-0.45	-0.41	-0.46	-0.33	-0.52	-0.35
N	-0.40	-0.34	-0.49	-0.27	-0.46	-0.30
D	-0.49	-0.43	-0.56	-0.41	-0.56	-0.36
C	-0.09	-0.21	0.03	-0.05	-0.17	-0.05
Q	-0.37	-0.30	-0.36	-0.34	-0.48	-0.32
E	-0.51	-0.41	-0.43	-0.30	-0.61	-0.39
G	-0.23	-0.37	-0.46	-0.37	-0.30	-0.33
H	-0.32	-0.26	-0.26	-0.30	-0.35	-0.25
I	-0.06	-0.08	0.26	0.09	-0.06	-0.07
L	-0.10	-0.18	0.02	0.04	-0.22	-0.13
K	-0.39	-0.45	-0.51	-0.35	-0.59	-0.32
M	-0.17	-0.25	-0.02	-0.10	-0.19	-0.18
F	-0.13	-0.11	0.05	-0.03	-0.13	-0.11
P	-0.56	-0.38	-0.56	-0.51	-0.42	-0.45
S	-0.37	-0.35	-0.41	-0.30	-0.48	-0.23
T	-0.34	-0.33	-0.28	-0.23	-0.40	-0.23
W	-0.17	-0.17	-0.09	-0.06	-0.12	-0.16
Y	-0.23	-0.11	-0.13	-0.06	-0.18	-0.15
V	-0.05	-0.14	0.19	0.14	-0.19	0.01

Table 1: Amyloid Effect (AE) Matrix

Position-specific ranking of amino acids

Using the AE matrix, if we sort the amino acids on every position, we get different rankings (but there are similarities).

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	V	I	C	L	F	M	W	G	Y	A	H	T	S	Q	K	N	R	D	E	P
2	I	F	Y	V	W	L	C	M	H	Q	A	T	N	S	G	P	R	E	D	K
3	I	V	F	C	L	M	W	Y	H	A	T	Q	S	E	R	G	N	K	D	P
4	V	I	L	F	C	W	Y	M	A	T	N	H	E	S	R	Q	K	G	D	P
5	I	W	F	C	Y	M	V	L	G	H	T	P	A	N	Q	S	R	D	K	E
6	V	C	I	F	L	Y	W	M	A	T	S	H	N	Q	K	G	R	D	E	P

Table 2: Ranking of amino acids on the 6 positions

Extremal values

The (positional) extremal values of the AE matrix show the amino acids that are the most and least connected to the amyloid property.

	1	2	3	4	5	6
ARGMAX	V	I	I	V	I	V
MAX	-0.047	-0.076	0.259	0.142	-0.064	0.008
ARGMIN	P	K	P	P	E	P
MIN	-0.559	-0.453	-0.562	-0.506	-0.605	-0.451

Table 3: Extremal amino acids from AE matrix

One could use these amino acids to transform a non-amyloid peptide into an amyloid one, with less amino acid replacement.

Amyloid and non-amyloid indicators

We wanted to construct simple rules, which indicate the amyloid or non-amyloid property of the hexapeptide in some cases.

We formulated the following problem.

Problem 1. *Are there any amino acids in proper positions, which guarantee that the hexapeptide would be amyloid?*

(Similarly to non-amyloid.)

Example 2. *The (P,P) amino acid pair on the (3,4) position pair is a non-amyloid indicator, if into the xxPPxx "incomplete" hexapeptide we change arbitrary amino acids with "x"s, the resulting hexapeptide would always be non-amyloid.*

The problem could be answered using the SVM linear predictor. Firstly we assume, that the SVM classifies without error, then we extend the result with SVM error quantities.

Perfect SVM: amyloid indicators

If the SVM is perfect, then the following "incomplete" hexapeptides would always result in amyloids with arbitrary substitution into the positions of "x"s':

'VIIVxx', 'VIIxIx', 'VIIxxV', 'VIxVIx', 'VxIVIx', 'VxIVxV', 'VxIxIV', 'VxxVIV', 'xI-IVIx', 'xIIVxV', 'xIIxIV', 'xxIVIV'

Here, one "incomplete" hexapeptide formulates a rule for 400 real hexapeptides. Similarly to these, if we could fix 4 amino acids, we could formulate many rules, altogether 3047.

But there is no rule with only 3 amino acids fixed.

Perfect SVM: non-amyloid indicators

If the SVM is perfect, then the following "incomplete" hexapeptides would always result in non-amyloid with arbitrary substitution into the positions of "x"s':

'PxPxxx', 'PxDxxx', 'xxPPxx', 'xxPDxx', 'xxPGxx', 'xxPKxx', 'xxPQxx', 'xxDPxx', 'xxDDxx', 'xxDGxx', 'xxDKxx', 'xxDQxx', 'xxKPxx', 'xxKDxx', 'xxNPxx', 'xxGPxx', 'xxRPxx', 'xxPxEx', 'xxPxKx', 'xxPxDx', 'xxDxEx', 'xxDxKx', 'xxDxDx', 'xxKxEx'

Here, one "incomplete" hexapeptide formulates a rule for 160.000 real hexapeptides. If we could fix 2 amino acids, these are the only rules, altogether 24.

But there is no rule with only 1 amino acid fixed.

Non-perfect SVM: correctness of indicators

The SVM predictor has the following measures for errors: $TPR = 0.747$, $TNR = 0.896$, $PPV = 0.804$, $NPV = 0.861$.

With these quantities, we could approximate the correctness of the previous rules.

For an amyloid indicator (e.g. VxIVIx) the rule always gives a positive (amyloid flag). Based on PPV, an amyloid rule is expected to be 80 percent accurate.

For a non-amyloid indicator (e.g. xxPPxx) the rule always gives a negative (non-amyloid flag). Based on NPV, a non-amyloid rule is expected to be 86 percent accurate.

Amyloid indicators on groups

We reformulated the previous problem because we were interested in the rules if we could replace the "x" only from a predefined group (not all 20 amino acids).

Problem 3. *Problem Are there any amino acids in proper positions, which guarantee for a group that the hexapeptide would be amyloid?*

Example 4. *Example The CxIWxx "incomplete" hexapeptide is an amyloid indicator for the GAST group, if we arbitrarily replace "x"s with one from GAST, the resulting hexapeptide would always be amyloid.*

Amino acid groups

We observed 3 normal amino acid groups and 2 artificial groups. [2]

The table includes (in this order) the name of the group, the members of the group, the least number of fixed amino acids required for a rule, the number of (minimal) rules.

Homogen class name	Class elements	Least k ind	N of ind
small nonpolar	GAST	3.0	323.0
hydrophobic	CVLIMPFYW	3.0	39.0
polar	DENQHKR	3.0	4.0
Artificial classes			
hydrophobic - {P}	CVLIMFYW	1.0	34.0
amino acids - {P}	QFYESNCMLIAHGWRKVT	3.0	4.0

Table 4: Amyloid indicators for groups

Examples for the normal groups

small nonpolar (first 10): VIIxxx IIIxxx VxIVxx VxIIxx VxILxx VxIFxx VxICxx VxIWxx VxVVxx VxVIxx

hydrophobic (all): VxIVxx VxIIxx VxILxx VxIFxx VxVVxx VxVIxx VxVLxx IxIVxx IxIIxx IxILxx IxVVxx IxVIxx CxIVxx CxIIxx CxILxx CxVVxx CxVIxx LxIVxx LxI-Ixx LxILxx LxVVxx LxVIxx FxIVxx FxIIxx FxVVxx MxIVxx MxIIxx MxVVxx xxIVIx xxIVxV xxIVxC xxIVxI xxIVxF xxIIxV xxIIxC xxILxV xxVVxV xxVVxC xxVIxV

polar (all): xxIVIx xxIVWx xxIIIx xxVVIx

Examples for artificial groups

hydrophobic - {P} (all): Vxxxxx Ixxxxx Cxxxxx Lxxxxx Fxxxxx Mxxxxx xIxxxx xFxxxx xYxxxx xVxxxx xWxxxx xLxxxx xxIxxx xxVxxx xxFxxx xxCxxx xxLxxx xxMxxx xxxVxx xxxIxx xxxLxx xxxFxx xxxCxx xxxWxx xxxIx xxxWx xxxFx xxxCx xxxYx xxxxxV xxxxxC xxxxxI xxxxxF xxxxxL

amino acids - {P} (all): xxIVIx xxIVWx xxIIIx xxVVIx

Conclusion

With the usage of a powerful linear predictor (SVM) we were able to extract useful information about the amyloid property of hexapeptides. The determined simple rules (e.g. xxPPxx is non-amyloid with every possible substitution) help chemists in the construction of peptides with desired properties. The anti-amyloid property of proline (P) was illustrated again with the amyloid and non-amyloid indicators.

References

- [1] Keresztes, L.; Szögi, E.; Varga, B.; Farkas, V.; Perczel, A.; Grolmusz, V. *The Budapest Amyloid Predictor and Its Applications*, *Biomolecules* 2021, 11, 500. <https://doi.org/10.3390/biom11040500>
- [2] Lesk, A. M. *Introduction to bioinformatics*. (2019).

Predicting the amyloid state by Support Vector Machines

Evelin Szögi

Institute of Mathematics, Eötvös Loránd University

szogievelin@student.elte.hu

Motivation

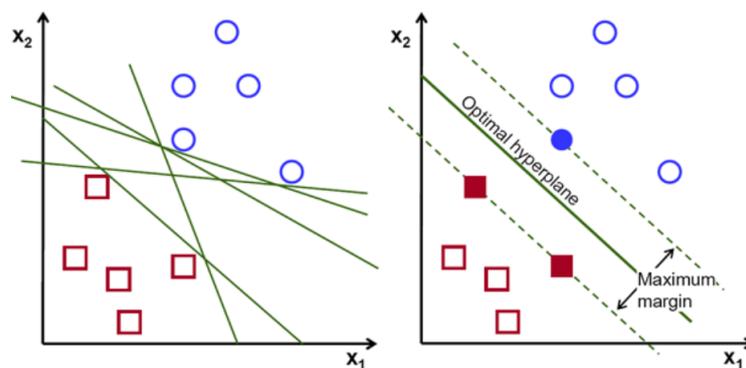
During the project, one of the main goals was to build a model, which can predict that a given peptide is amyloid or not. We don't have any databases which contains other external factors, only the amino acid sequence and amyloid - non-amyloid labels. Thus the model has to use only the amino acid sequences. As Machine Learning (ML) is a very powerful tool, we used ML techniques, more precisely Support Vector Machines (SVM). As SVM is one of the simplest and most interpretable ML tools, the main goal was to build an SVM model with high prediction accuracy and other good metrics and to extract new information from the model about the ability to form an amyloid state.

Support Vector Machines

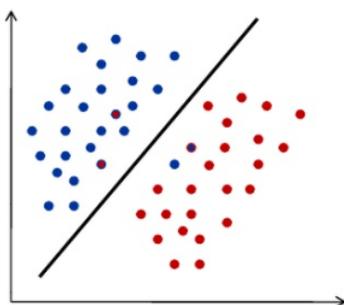
In machine learning, Support Vector Machines are usually used for classification tasks. In the 2-dimensional space, the problem is the following:

N points are given in \mathbb{R}^2 : $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N \in \mathbb{R}^2$. Every point has a color $c_i \in \{\text{blue}, \text{red}\}$ $i = 1, \dots, N$. We want to find a line separating the points with different colors.

If there exists a separating line, infinity number of separating lines exist. Therefore, we want to find the best separator, that is, the line with the widest margin.



It can be shown that when searching for the best separator, one has to solve an optimization problem (a quadratic convex function has to be minimized). In some cases, the two class of points are not separable, so it is not expected to separate the points correctly, some of them can lie on the wrong side of the separator, but the number of misclassified points are controlled.



In \mathbb{R}^n , we generalize the mentioned ideas for vectors and hyperplanes. To find the optimal separating hyperplane, one has to solve the following convex quadratic optimization problem (QP):

$$\begin{aligned} \min_{\mathbf{w}, b, \xi} \quad & \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^N \xi_i \\ \text{subject to} \quad & y_i(\mathbf{w}\mathbf{x}_i + b) \geq 1 - \xi_i \text{ for } i = 1, \dots, N \\ & \xi_i \geq 0 \text{ for } i = 1, \dots, N \end{aligned}$$

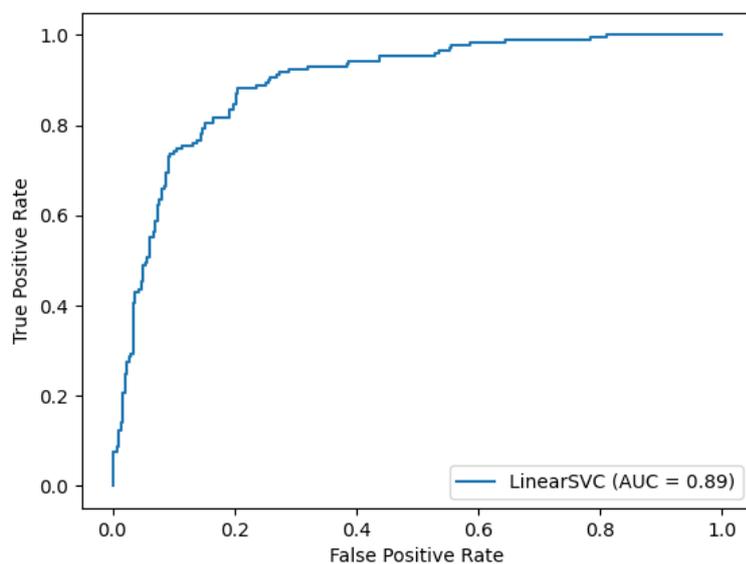
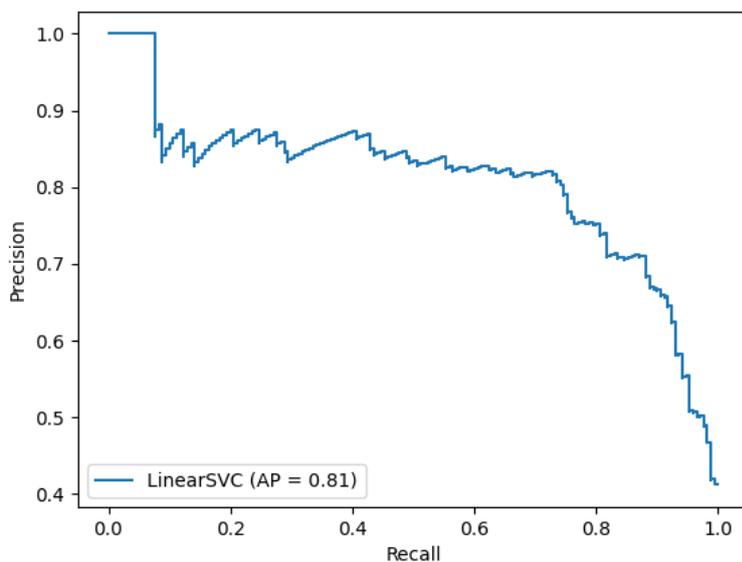
Database, representation

We got our data from the Waltz database, which contains 1415 sequences, all of them are of length six. The database also contains the amyloid - non-amyloid labels. It is available here: <http://waltzdb.switchlab.org/sequences>. If we want to use SVM, we have to represent the hexapeptides as real vectors. This can be done by using another database. AAindex is a database of numerical indices representing various physicochemical and biochemical properties of amino acids. Using AAindex database, we can represent an amino acid as a real vector with more than five hundred coordinates by collecting all the numerical values from the database. This way, it is possible to represent hexapeptides as vectors: we just have to concatenate the corresponding amino acid vector representations. So, all hexapeptides are represented by real vectors with more than three thousand coordinates. AAindex is available here: <https://www.genome.jp/aaindex/>.

Classification by using SVM

We build support vector machine models with different hyperparameters. Using the best model, $84.15 \pm 3.31\%$ of the test data was classified correctly with 95% confidence interval. The table below shows the true positive rate, true negative rate, the positive predicted value and the negative predicted value.

TPR TNR PPV NPV
0.754 0.895 0.803 0.865



Feature selection

We have more than 3000 features and the normal vector of the separating hyperplane has many zero coordinates. This suggests us, maybe there are redundant coordinates in

the vector representation. Therefore, we wanted to drop the features which don't give us major information and select the important ones. It can be done by using feature selection methods.

One simple feature selection algorithm is the following:

- Find the best separating hyperplane, and let w denote its normal vector;
- Order the features in descending order with respect to $|w_i|$ ($i=1,\dots,3318$);
- Drop the features at the end of the order as these are possibly redundant features.

This way, we get the features which are indeed important in the amyloid - non-amyloid classification problem. Using the algorithm, we managed to find 17 features, which is much more than 3000. Using only these features, the classification accuracy is still high, 80%. There are 9 features of the 17 features from which the classification accuracy is 78%.

Next steps

One of the next steps is improving the SVM model. For this purpose, active learning techniques could be used. It's a special case of machine learning. It's used, when there are lots of unlabeled data but it's expensive to label them. In active learning, a learning algorithm can interactively query a user to label new data points. The algorithm will ask for the most uncertain points labels. We would also like to make predictions for longer peptides using different techniques, too.

Sliding windows in the Protein Data Bank: amyloid-forming propensity of prefixes and suffixes of secondary structures

Kristóf Takács, Vince Grolmusz

Department of Computer Science, Eötvös Loránd University, Budapest

takacs.kristof.elte@gmail.com, grolmusz@pitgroup.org

The main aim of this work was to test and validate the Budapest Amyloid Predictor [3] on real-life data by creating several databases from the Protein Data Bank [1] and using the entries of these databases as input of the predictor. By examining these datasets, interesting results regarding the frequencies of hexapeptides predicted as amyloid-forming were observed which support the assumption that the Budapest Amyloid Predictor probably can be applied as a useful tool in determining the amyloid-forming propensity of hexapeptides.

Principal idea: sliding windows

In a 2015 paper by Família et al. [2] which describes a neural network model (APPNN) applicable for the problem regarding the prediction of the amyloid-forming propensity of proteins, numerous statistical values were assigned to each amino acid (e.g., based on their incidence in α -helices and β -sheets). Our idea was to apply the same method except not for individual amino acids but peptides consisting of six amino acids, i.e. hexapeptides.

The execution of this idea included a so-called "sliding window" with a length of six amino acids. This window was pushed through every α -helix and β -sheet section as well as every section with a neither α -helix nor β -sheet secondary structure (referred to in this work as "other" secondary structure) of the entries in the 30% homology filtered Protein Data Bank (PDB), creating three new databases from the hexapeptides which appear in α -helix, β -sheet or "other" sections, respectively. Following this step, different statistical values were obtained from these databases, e.g., the frequency of hexapeptides in different secondary structures.

In Figure 1, there is an illustration as an example of this method: suppose that in an entry of the homology filtered PDB, there is an α -helix section which consists of the amino acids between the two bolded lines, i.e. ***GGKQALETVVAIAS***. The first hexapeptide observed by the sliding window is the peptide of the first six amino acids in the section, i.e. *GGKQAL*. After storing this peptide in a list, the window is moved by one amino acid to the right, meaning that the next hexapeptide to be stored in a list is the one that expands from the 2nd amino acid to the 7th amino acid of the section, i.e. *GKQALE*. This method is applied repeatedly until the last six amino acids of the section (in this case, *VVAIAS*) are reached. This sliding window approach was used for every section with the given secondary structure in the homology filtered PDB, creating three new datasets, one for each secondary structure.

After assembling these new datasets, it was possible to apply the SVM model of the Budapest Amyloid Predictor to the sets of hexapeptides belonging to each secondary

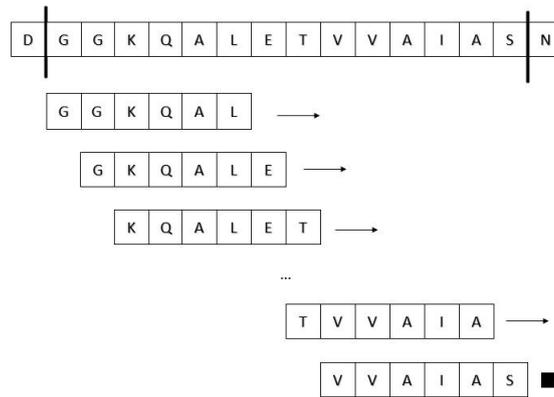


Figure 1: An illustration of the hexapeptides derived from the application of a sliding window.

structure. These input databases consisted of approx. 1.62 million, 308,000 and 950,000 different sequences for α -helix, β -sheet and "other" secondary structure sections. From the output of the SVM model, the prediction labels ("amyloid" / "not amyloid") and the numeric values (SVM score) considered in the classification process could also be used in further examination.

Based on these data, the histograms of the SVM scores in each secondary structure dataset were created: these diagrams closely resembled bell curves (see Figure 2), leading to the assumption that the scores which belong to the same secondary structure may be originated from a normal distribution.

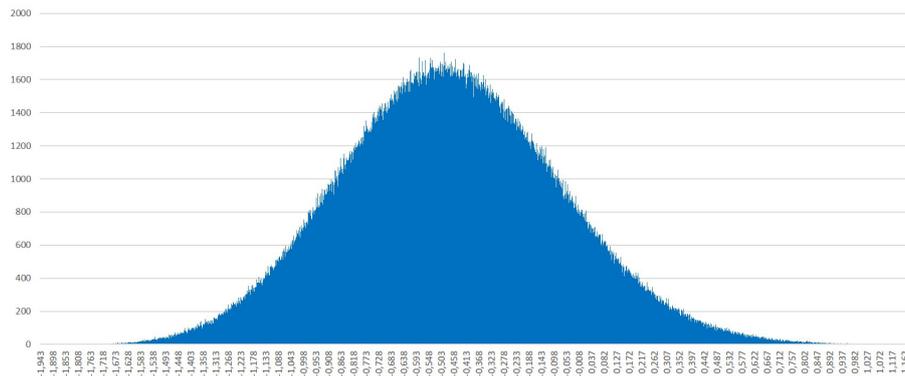


Figure 2: Histogram of α -helix hexapeptide SVM scores.

The output of the SVM scores implies that on average, hexapeptides from an α -helix section are less amyloid-like than ones from a β -sheet section, but more amyloid-like than hexapeptides from a section with a neither α -helix nor β -sheet secondary structure.

Prefixes and suffixes of secondary structures

A similar approach using sliding windows was also applied to the end segments of the different secondary structure sections: for every section belonging to a secondary structure, 14 hexapeptides were collected in the following way (see Figure 3):

For a prefix, the sliding window starts at the same position as described before, i.e. the first six amino acids of the given section. After that, the sliding window is moved by one amino acid to the *left*; this step is repeated until the window contains the hexapeptide which directly precedes the given section. During the process, 7 hexapeptide can be seen in the sliding window; these hexapeptides are stored in 7 different lists, according to the number of their amino acids which do not belong to the original secondary structure section (denoted by k on the figure). This method is executed for every section with a given secondary structure, yielding 7 databases for every secondary structure. (A very similar approach can be applied for suffixes, starting with the last six amino acids of a section, then moving the window by one amino acid to the right, etc.)

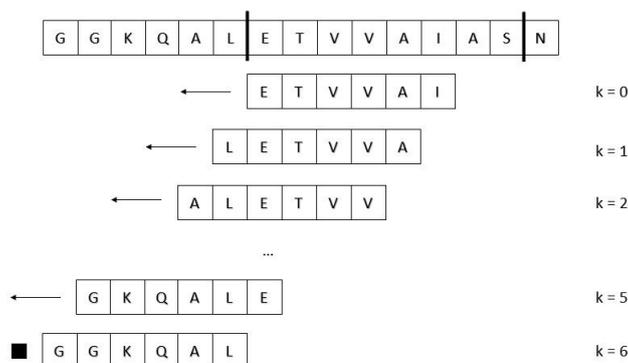


Figure 3: An illustration of the hexapeptides derived from the application of a sliding window for the prefix of a secondary structure section.

After creating these 14 datasets (7-7 for prefixes and suffixes) for each secondary structure, it was possible to determine the ratio of amyloid-like hexapeptides in each dataset with regard to their k value (the number of their amino acids outside the examined section). E.g., as it can be seen in Figure 4, the process of dropping of the ratio of amyloid-like hexapeptides in β -sheet prefixes (according to the SVM model) as k increases is quite fast: from $k = 0$ to $k = 3$, the ratio decreases from about 33% to 5%, but for greater k values, the ratio stays about at the same level. It is quite interesting that more or less the same diagram can be generated if β -sheet suffixes are considered (see Figure 5).

Similar results were achieved for other secondary structures as well as when stricter conditions were applied during the process of filtering the examined sections.

Conclusion

Applying the SVM classifier (Budapest Amyloid Predictor) developed by members of our research group for real-life protein sequence data from the Protein Data Bank, promising results were achieved, mainly in line with preliminary expectations. As a conclusion,

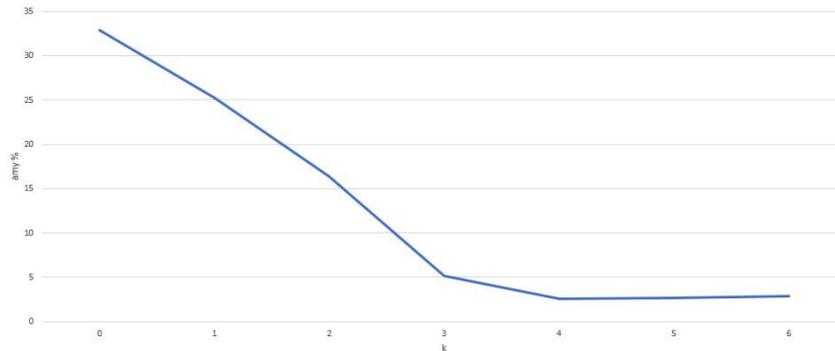


Figure 4: Ratio of hexapeptides predicted as amyloid-like in β -sheet prefixes depending on the number k of amino acids outside the given section.

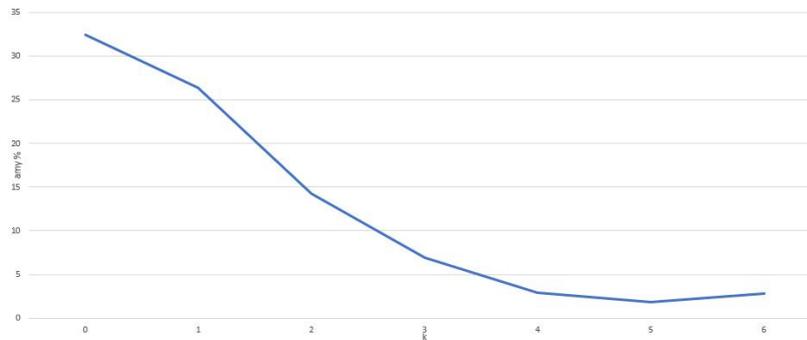


Figure 5: Ratio of hexapeptides predicted as amyloid-like in β -sheet suffixes depending on the number k of amino acids outside the given section.

it can be strongly assumed the Budapest Amyloid Predictor can be considered as a useful tool in determining the amyloid-forming propensity of hexapeptides.

References

- [1] **Berman, H.M., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T.N., Weissig, H., Shindyalov, I.N. and Bourne, P.E.**, The Protein Data Bank, *Nucleic Acids Research*, **28** (2000), 235–242. <http://dx.doi.org/10.1093/nar/28.1.235>
- [2] **Família, C., Dennison, S.R., Quintas, A., Phoenix, D.A.**, Prediction of Peptide and Protein Propensity for Amyloid Formation, *PLOS ONE*, **8** (2015). <https://doi.org/10.1371/journal.pone.0134679>
- [3] **Keresztes, L., Szögi, E., Varga, B., Farkas, V., Perczel, A. and Grolmusz, V.**, The Budapest Amyloid Predictor and its Applications, *Biomolecules*, **4** (2021), 500. <https://doi.org/10.3390/biom11040500>

Pathfinding in the hexapeptide-graph: through the amyloid and non-amyloid nodes

Bálint Varga

Department of Computer Science, ELTE

balorkany@pitgroup.org

Background

Amyloid formation has long been associated with several diseases, such as Parkinson's disease, Alzheimer's disease and Creutzfeldt-Jakob's disease, to name a few. Therefore understanding amyloid formation could be advantageous in understanding these associated diseases, but regardless of the benefits, our knowledge of the topic is limited.

Using the Support Vector Machine (SVM) [1] method and an available database of characterized hexapeptides, the WALTZ-DB 2.0 [5], our research group created a prediction algorithm for hexapeptides [3], which gave us an opportunity to define the hexapeptide-graph:

Definition 1 *Hexapeptide-graph*

Let $G = (V, E)$ a simple, undirected graph, where

- $v \in V$ are all the 20^6 possible hexapeptide.
- For $v_1, v_2 \in V$, $\{v_1, v_2\} \in E \iff \text{Hamm}(v_1, v_2) = 1$, where $\text{Hamm}()$ is the Hamming distance.
- There is an $\text{Ami} : V \rightarrow \{0, 1\}$ label on each node, based on our SVM predictor describing the amyloidicity of the underlying hexapeptide.

On this hexapeptide-graph, to support the team who in practice is making the synthesis, we had two tasks:

Given a Start and Stop hexapeptide:

- Find all possible paths between a Start and Stop hexapeptide with length k , using only amyloid or non-amyloid nodes.
- Find a shortest path between a Start and Stop hexapeptide, while using only amyloid or non-amyloid nodes and shortest is minimizing for
 - number of point-mutations during the path (Hamming distance),
 - a weighting of the amino acid alphabet (referring to the difficulty of synthesis),
 - the actual number of nodes.

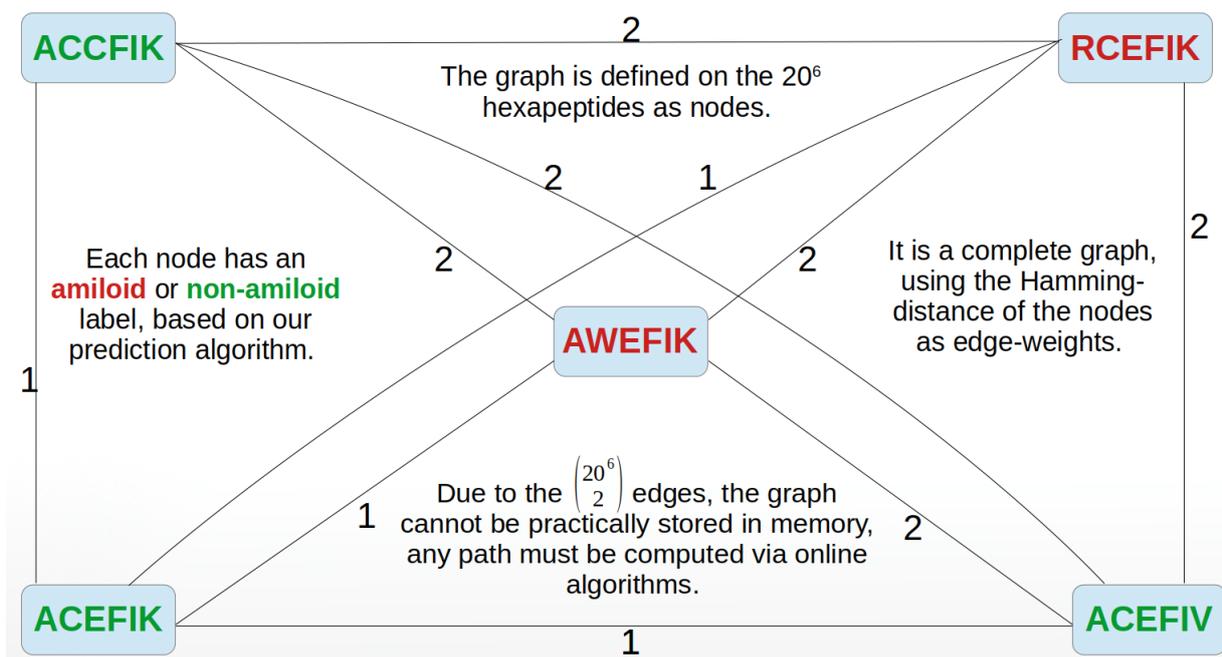


Figure 1: A 5 node subgraph of the hexapeptide graph. Green: Amyloid, Red color: non-amyloid

All paths with length k

For the first problem, we have used a depth-first search (DFS) based backtracking algorithm [6] with a pruning condition. The DFS base is due to practical concerns: with typically longer path-segments and fewer branches, it favors Memory in the Memory-Time trade off.

The idea for the pruning: suppose we start at t and s is the stopping node. During the run of our DFS, we are at a certain node c with a corresponding path-segment ps . If

$$Hamm(c, s) + |ps| > k, \text{ then}$$

due to the construction of the graph, ps cannot be finished so that the resulting path has length k .

Here is a sample using LATVYV as starting and VQIVYK as stopping node for 6-long paths:

```
LATVYV PATVYV PQT VYV PQT VYK PQIVYK VQIVYK
LATVYV PATVYV PATVYK PQT VYK PQIVYK VQIVYK
LATVYV LARVYV LARVYK VARVYK VQRVYK VQIVYK
LATVYV LARVYV LARVYK LQRVYK VQRVYK VQIVYK
```

Shortest path

For the second problem, in this special case we have two distinct distance metric on the nodes:

- the usual graph-metric of how many edges does it take to get from t to s , and
- a global distance metric of Hamming distance of the underlying hexapeptides.

This makes the problem similar to path finding on road networks, where the connecting roads and the physical distance defines the problem. Borrowing from the analogue problem, we have chosen an extension of Dijkstra’s algorithm, the A^* [2] algorithm, because it is often used in this situation.

A^* is an informed search algorithm, meaning it incorporates a ‘sense of direction’, a heuristic of how different the current path-segment from an abstracted strait-line. With this heuristic incorporated, the algorithm guaranteed to find an optimal path without processing any node more than Dijkstra’s algorithm, but in practice, it is usually considerably better.

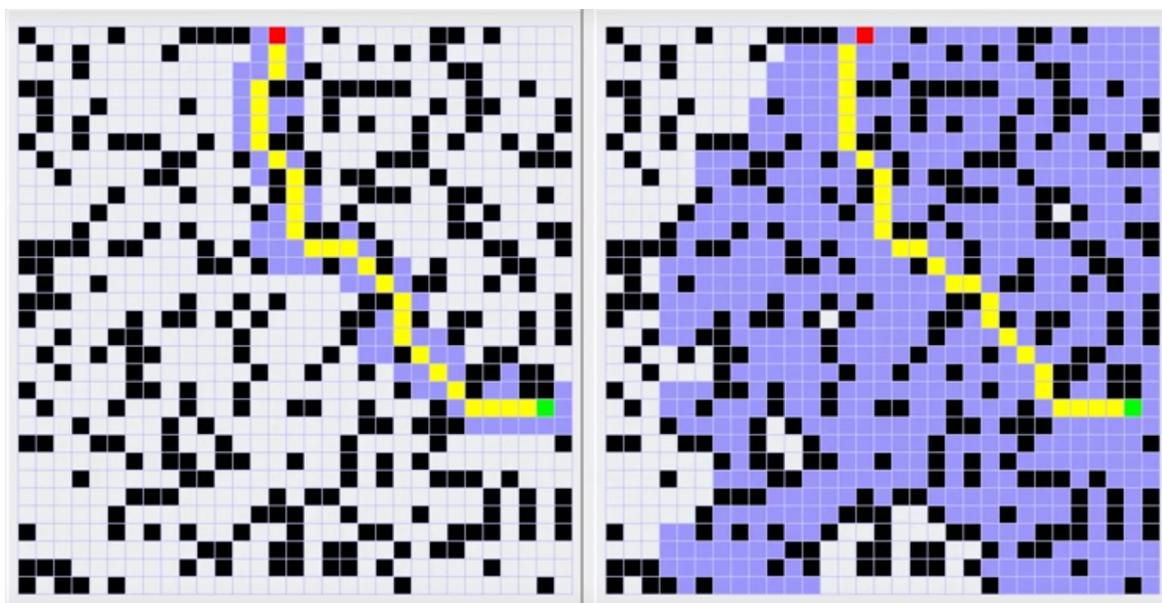


Figure 2: A simulation between A^* algorithm (left) and *Dijkstra* algorithm (right). Green: start position, Red: stop position, Yellow: the proposed path, Blue: processed nodes [4]

In our case, the cost-function had to incorporate a weighting of the amino acid alphabet, mirroring the difficulty of synthesis. In the end, our custom cost-function took the form of

$$c(ps) = |ps| + Hamm(ps[-1], s) + \sum_{n \in ps} W_{alphabet}(n), \text{ where}$$

ps is the actual path-segment, $ps[-1]$ is the last node of the path-segment, $Hamm()$ is the Hamming distance and $W_{alphabet}$ is the weighting of the amino acid alphabet.

Here is a sample using LATVYV as starting and VQIVYK as stopping node:

LATVYV LATVYK AATVYK AQTVYK AQIVYK VQIVYK

References

- [1] **Evgeniou T., Pontil M.** *Support Vector Machines: Theory and Applications.*, Machine Learning and Its Applications. ACAI 1999. Lecture Notes in Computer Science, vol 2049.
- [2] **Hart, P., Nilsson, N., Raphael, B.** *A Formal Basis for the Heuristic Determination of Minimum Cost Paths.*, IEEE Transactions on Systems Science and Cybernetics, 4(2), 100–107. <https://doi.org/10.1109/tssc.1968.300136>
- [3] **Keresztes, L.; Szögi, E.; Varga, B.; Farkas, V.; Perczel, A.; Grolmusz, V.** *The Budapest Amyloid Predictor and Its Applications.*, Biomolecules 2021, 11, 500. <https://doi.org/10.3390/biom11040500>
- [4] **Kevin Wang** *Compare A^* with Dijkstra algorithm*, <https://github.com/kevinwang1975/PathFinder>
- [5] **Louros, N. et al.** *WALTZ-DB 2.0: an updated database containing structural information of experimentally determined amyloid-forming peptides.*, Nucleic Acids Res, 48 (D1), D389–D393. <https://doi.org/10.1093/nar/gkz758>
- [6] **Tarjan, R.** *Depth-First Search and Linear Graph Algorithms.*, SIAM J. Comput. 1 (1972): 146-160.

Section:

Selected Topics

Organizer: Erzsébet Csuhaj-Varjú

Contributions:

- Zsófia Erdei, Melinda Tóth and István Bozó: Targeted static fault localization in Erlang programs
- Beka Grdzelishvili and Viktória Zsók: Design and Implementation of Digital Image Processing in Functional Programming
- Jianhao Li and Viktória Zsók: Actor Model based Distributed Communication in Golang
- Pramod Kumar Sethy: Notes on P systems versus R systems
- Gabriella Tóth and Máté Tejfel: Error detection and analysis of PSA structured P4 programs



Targeted static fault localization in Erlang programs¹

Zsófia Erdei, Melinda Tóth, István Bozó ELTE, Eötvös Loránd University,

Faculty of Informatics

`zsanart@inf.elte.hu`, `toth_m@inf.elte.hu`, `bozo_i@inf.elte.hu`

Abstract

Static source code analysis techniques may help the programmers in various tasks: code comprehension, testing, debugging, etc. They often need to reproduce executions that result in faulty behaviour. Program analysis techniques with symbolic execution can help to solve this task. In this paper we propose a method to select an appropriate execution path from the static control-flow graph that may lead to a given runtime error in Erlang software. We build our tool on the RefactorErl static analyser framework.

1 Introduction

Fault localization is the act of identifying the locations of faults in a program. Even when bugs in software are discovered due to some faulty behavior (e.g. a runtime error occurs), finding the location of the fault is a non-trivial task. Error detection mechanisms are vital for building highly reliable systems. Fault localization is one of the most time consuming, and expensive part of software development and maintenance. Given the size and complexity of large-scale software systems today, manual fault localization becomes more and more futile, so effective automatic methods are needed.

In a concrete execution, a program is evaluated on a specific input and a single control-flow path is explored. Symbolic execution [4] uses unknown symbolic variables in evaluation, allowing to simultaneously explore multiple paths that a program could take under different inputs.

We can use symbolic execution to help us in fault localization. We target to find an execution path in the program, the "error path", that may result in a runtime error in a given point of the program. Thus we build a direct symbolic execution engine for a given execution path in the Erlang programs based on the RefactorErl framework [7]. We are using the SMT solver of Z3 [2] to solve the constraints that we gather during our analysis.

The main goal of our work is to provide an algorithm and a tool that helps Erlang developers to reproduce a faulty behaviour that results in a runtime error. In this paper, we present the algorithm based on the static analyses functionalities already defined by RefactorErl.

2 Defining the problem

In this section we will introduce the problem which can be described as the line-reachability problem: given a target expression or line in the program, we want to find a realizable

¹The research has been supported by the European Union, co-financed by the European Social Fund (EFOP-3.6.3-VEKOP-16- 2017-00002). The research is part of the "Application Domain Specific Highly Reliable IT Solutions" project that has been implemented with the support provided from the National Research, Development and Innovation Fund of Hungary, financed under the Thematic Excellence Programme no. 2020-4.1.1.-TKP2020 (National Challenges Subprogramme) funding scheme.

path to that point in the control-flow graph, if it exists. This problem is the equivalent of the very general problem of finding a path that causes the program to enter a particular state, that can be especially useful for debugging or regression testing over a program. For example, if we know that an error may occur at a given line in the code, knowing the path - and so the conditions and input values leading to the fault, we may have an easier time fixing it.

```

1 -module(example1).
2 -export([foo/2]).
3 foo(A, B) ->
4   if
5     A rem 2 == 1 ->
6     C = A + B,
7     A / C;
8   true ->
9     C = A - B,
10    A / C
11 end.
```

For the sake of simplicity, a division by zero will illustrate the bug we intend to find. Consider the following example Erlang code on the left, and suppose during execution we encountered a division by zero error on line 10. Let us assume our goal is to determine with which inputs do we reach the line containing the possible fault. Exploring the control-flow graph built from the function we can see the graph branches off at the if expression. To reach the target line in the execution path the first condition has to be false and the second condition has to be true (in this case the second condition is the default "true"). Our set of conditions will contain the negation of the first condition of the if expression, and the condition "true". Evaluating this set of conditions $\neg(A \bmod 2 = 1)$ and *true* or simply $A \bmod 2 = 0$, we can find that the error can occur only if the variable *A* is even. This is of course true, but it is not enough information in

itself to highlight the cause of the error. As we can see the error originates from the value of the variable *C* being equal to zero, so we should add this to our set of conditions: $C = 0$

This condition for the variable *C* and the previously established condition for *A* determine the conditions leading to the error, however, they are not enough on their own to find the input values possibly leading to the error. It is easy to see, that the value of *C* will be zero only if $A = -B$, but our set of conditions does not say anything of the value of *B*. Our set is not yet complete, since the variable *C* is not an input parameter of the function, we have to find the match expression that assigns value to the variable. Using this as our next condition the following can be established:

$$\neg(A \bmod 2 = 1), C = 0, C = A + B$$

Given this set of conditions, we have constraints on both input parameters *A* and *B*. Evaluating these with the use of an SMT solver we can find values that satisfy these conditions leading to faulty behaviour.

3 Defining the algorithm

In this section, we present our algorithm for symbolic execution to find a path from the program entry point to the specified target expression. This algorithm can be used to determine possible values of input parameters needed to reach the target line, and also presents a set of conditions for these parameters that need to be satisfied. The algorithm uses a form of symbolic backward execution called call-chain backward symbolic execution [5], a variant of symbolic execution which uses a combination of the traditional forward symbolic execution and symbolic backward execution. While inside each function the exploration is done with forward symbolic execution, the analysis follows the call-chain backwards from the target point to the entry point of the program.

The exploration starts at the target expression. First, we determine the path from the function containing the target to the target expression itself. For this, we use the control-flow graph built from the function and explore it to find a feasible path to the target node while at every branch collecting path conditions along the path. The CFG of the function is traversed in a breadth-first manner, and the first found path is returned. This gives us a set of expression nodes in the Semantic Program Graph, that can be used to determine the initial set of conditions. Once a path has been found we need to check the set of conditions for satisfiability. On failure, we need to discard the path and find another one. Using this method we can determine a valid path from the function entry point to the target. After this, we determine the callers of the function, and recursively repeat this process until we are able to find a path from the program entry point to the target expression.

Our initial set of conditions is based on the conditions collected from the branches of the if expressions. When we are walking a selected path on the control-flow graph, we need to add the conditions that were met along the path, since those conditions had to be met to get to the selected target point using this path. However, this will not be enough. In Erlang the branches of an if expression are scanned sequentially until a guard sequence that evaluates to true is found. This means that if we found that in an if expression the second condition was met, we also know, that the first condition evaluated to false. For this reason, we also need to add the negate of every previous condition that have not been met to our set.

While building the set of conditions, we also need to keep track of the variables in the conditions. While exploring the control-flow graph, every semantic variable node when first encountered is saved in a map data structure with the original name of the variable. When later a different variable with the same name is encountered, the new variable will be renamed, ensuring that every instance of a variable in the set of constraints will indicate the same semantic variable.

4 Related Work

Symbolic execution is not a new topic in the Erlang ecosystem. Formal [8] and informal [3] definitions were published with the aim of program verification. Our research is not focusing on verification, we aim to support debugging processes of the Erlang developers through the RefactorErl framework.

Besides verification, symbolic execution is used for testing purposes in the context of Erlang. CutErl [6] introduces a concolic testing framework for Erlang through a dynamic symbolic execution framework. Our approach is similar to the mechanism of CutEr. Both methods are collecting symbolic constraints and evaluate them with the Z3 solver. However, our tool uses a full static approach and calculates the constraints on execution paths generated from the static control-flow graph and traverses it backwards. CutEr uses a concrete execution to perform a forward traversal.

The authors of [1] present a work on runtime error detection based on symbolic execution. They are transforming Erlang programs to Prolog facts and provide an interpreter to evaluate them on symbolic input data. Their analysis reports input patterns that lead to runtime errors within a given bound. Our analysis works in reverse order. It takes an occurred runtime error as input and searches for execution paths that may lead to the

faulty behaviour.

A generic algorithm for call-chain backward execution (CCBSE) was presented in [5] which in combination with any forward search strategy resulted in an efficient way to solve the line-reachability problem. In this algorithm, the main difference to the traditional symbolic backward execution is that, while it follows the call-chain backwards from the target point, inside each function the exploration is based on a traditional forward symbolic execution. For CCBSE, the availability of the inter-procedural control-flow graph is a crucial requirement, which is described as a disadvantage for the reason that constructing such a graph can be quite challenging in practice. However, RefactorErl already includes this functionality making the algorithm a favourable option for us to use it.

5 Conclusion and Future Work

Locating the sources of a runtime error is an everyday task of an Erlang developer. Dynamic and static tools could provide help in this task. In this paper, we proposed a method based on static analysis of Erlang programs to identify execution paths that may lead to a given runtime error. We use the control-flow graph of RefactorErl and apply dynamic backward symbolic execution on it to gather the constraints of the execution. We use the Z3 SMT solver to decide the applicability of a path and calculate possible input values for real execution.

The current implementation works only a subset of Erlang. Thus our main goal in the future is to improve our implementation and extend the language coverage. After this, we would like to improve the execution path selection algorithm based on the tested use cases.

References

- [1] Emanuele De Angelis, Fabio Fioravanti, Adrián Palacios, Alberto Pettorossi, and Maurizio Proietti. Bounded symbolic execution for runtime error detection of erlang programs. In Temesghen Kahsai and Germán Vidal, editors, *Proceedings 5th Workshop on Horn Clauses for Verification and Synthesis, HCVS 2018, Oxford, UK, 13th July 2018*, volume 278 of *EPTCS*, pages 19–26, 2018.
- [2] Leonardo de Moura and Nikolaj Bjørner. Z3: An efficient smt solver. In C. R. Ramakrishnan and Jakob Rehof, editors, *Tools and Algorithms for the Construction and Analysis of Systems*, pages 337–340, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg.
- [3] Clara Benac Earle. Symbolic program execution using the erlang verification tool. In María Alpuente, editor, *9th International Workshop on Functional and Logic Programming, WFLP'2000, Benicassim, Spain, September 28-30, 2000*, pages 42–55, 2000.
- [4] James C. King. Symbolic execution and program testing. *Commun. ACM*, 19(7):385–394, July 1976.
- [5] Kin-Keung Ma, Khoo Yit Phang, Jeffrey S. Foster, and Michael Hicks. Directed symbolic execution. In Eran Yahav, editor, *Static Analysis*, pages 95–111, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.
- [6] Kostis Sagonas. A cuterl tool. talk at erlang factory 2016, 2016. [Accessed: May, 2021].
- [7] M. Tóth and I. Bozó. Static analysis of complex software systems implemented in erlang. Central European Functional Programming Summer School – Fourth Summer School, CEFP 2011, Revisited Selected Lectures, Lecture Notes in Computer Science (LNCS), Vol. 7241, pp. 451-514, Springer-Verlag, ISSN: 0302-9743, 2012.
- [8] Germán Vidal. Towards symbolic execution in erlang. In Andrei Voronkov and Irina Virbitskaite, editors, *Perspectives of System Informatics*, pages 351–360, Berlin, Heidelberg, 2015. Springer Berlin Heidelberg.

Design and Implementation of Digital Image Processing in Functional Programming

Beka Grdzelishvili, Viktória Zsók

Department of Programming Languages and Compilers
Faculty of Informatics
Eötvös Loránd University
bekagrzelishvili0@gmail.com, zsv@inf.elte.hu

1 Project background

Digital image processing [2, 4] deals with the manipulation of digital images. It is a subfield of signals and systems, but the focus is particularly on images. The aim of this project is to design and build a program, which can process digital images using pure functional programming language. The input of the application is a digital image, which is processed using various algorithms, and it creates a modified image as an output. Digital Image Processing is widely used and it is popular among photographers, designers, graphic artists, and other artists. The most famous example application is Adobe Photoshop [1], which is used for image editing, creating compositions, and adding affects.

2 The PPM image format

The implemented application [3] gets a digital image as input and converts it into appropriate data structure for future modifications. As an image is described with several attributes, a new structure is created to store those attributes together and efficiently move data through the processing pipeline. The most important image attribute is the *pixel*, a list of RGB `Pixel` records. Each RGB pixel contains 3 integer values, which are the intensity of following colors: Red, Green and Blue. These 3 integers can be used to represent any color, while the collection of RGB pixels can be used to describe any image.

Digital images can be stored in various file formats, like JPG, PNG, BMP, PPM, RAW and etc. Each file format has its own advantages and disadvantages. After several experiments and tests, the PPM [5] image format was selected for this project. The PPM (*Portable Pixel Map*) format is the lowest common denominator color image file format. It is not famous for efficiency, as it is a very simple format, and it does not support image compression. That is why storing large images in PPM takes much more memory than in other formats. However, the simple structure makes it easier to read and process data stored in PPM image, it does not require any decompression algorithms, and the pixel data stays unchanged during I/O making it appropriate for the project.

The application reads data from PPM image file, converts it into an `Image` record, and after applying the desired processing effects, the output is written again in a PPM image file. Even though currently only PPM image I/O modules were implemented, it can be easily extended to include other formats as well.

3 Image processing

Image processing starts in the `Processor` module. The image is read using the `loadFromPPM` function, and after modifications, the output is generated using the `saveToPPM` function. Both functions are part of the `PPM.IO` module, which is responsible for handling the PPM format [5]. The image modification is done with pixel-wise processors, i.e. image filters. The filters iterate through the pixel list of the image, and they alter each RGB pixel record individually. The `Filters.Filter` module contains the `applyPixelFilter` function, which is responsible for processing the image according to the given filter. Each filter is a separate function, that takes the `Pixel` record, and it returns a new, reformed `Pixel` record.



Figure 1: Base image – before modifications.

The example image, Figure 1, is used to show visual changes on the image. It is a plain PPM image with 272190 pixels (645x422). Each pixel is described with 3 integers, encoding the RGB colors with the max value of 255. In total, the input pixel stream contains 816570 integers.

3.1 Grayscale filters

The first processing unit is the grayscale filter. It transforms the colored image into a black and white picture. The colors are lost and everything is gray, but of course the objects on the image are still distinguishable. The bright colors are changed with lighter tone of gray than the dark colors, see Figure 2.



Figure 2: Simple, weighted and threshold grayscale images.

Unlike RGB colors, pixels in a grayscale image are represented with a single number, indicating the intensity of gray. Therefore, each RGB value should be converted into one number. As values in RGB already represent color intensity, we can take the average of

all three as a grayscale value. This method is a simple conversion, which indeed gives us a black and white image. However, the number representation does not accurately match our eyes' perception. Colors can be close to black in RGB color space, but for us, it might be brighter. The weighted grayscale filter takes this fact into account to produce smoother grayscale images. In this filter intensities of Red, Green and Blue are not considered equal and a weighted average is calculated to compensate for the human perception.

The next filter is the threshold black and white filter that converts RGB colors of each pixel into grayscale. Unlike the previous filters, instead of different variants of gray, the result is either fully black or fully white pixel. The filter takes the threshold value and it compares to the grayscale value of the pixel. If the grayscale value is greater than the threshold, it is considered to be black; if not, it is changed with white pixel. The resulting picture may not look beautiful, but it can be useful to spot small details of the image.

3.2 Selective black and white filters

Another type of black and white filter is the selective filter. Such filters are popular in photo editor programs, they allow you to select one or more colors that you want to keep and everything else is turned into black and white, see Figure 3. Converting unmarked pixels into grayscale can be done with above mentioned filters, but before that, it is needed to mark pixels that are close to users' specified color. For that, we require to calculate the distance between pixels color and the specified color, and if its absolute value is less than the predefined threshold, it should stay unchanged.



Figure 3: Selective grayscale filter omitting the yellow color.

3.3 Tint and light filters

The next filter is used to modify the color palette of an image. Tint filter is used to increase or decrease specific color's intensity. For example, increasing the intensity of red in each RGB pixel, can make an image more 'reddish'. Colors are often used to put extra information in the image; for example, if the color palette is closer to blue, it feels like that photo was taken in a cold environment, while more 'yellowish' colors give the feeling of warmth, see Figure 4.

The last type of pixel filters presented here also interferes with the color palette, but in a bit different way than the tint filter. If an image is too dark, the decreasing intensity of each color can result in a brighter image, as in Figure 5. Likewise, increasing the intensity



Figure 4: Images tinted with different colors.

of each RGB value can give us a darker image. The brightness filter uses this method to change the light in the image.



Figure 5: Brighter and darker variations of the base image.

4 Summary

The image processing application was developed using functional programming. Several image processing filters were integrated into the application. During testing and development, functional programming was found to be suited for the image processing application purposes. With the help of the functional paradigm, the code's design remained small and well-structured during the development. High-order functions, currying and lazy evaluation made processing more efficient, especially chaining different filters and effects.

References

- [1] **Adobe Photoshop**, www.adobe.com.
- [2] **Digital Image Processing Course**, <https://www.ft.unicamp.br/docentes/magic/khoros/>.
- [3] **Digital Image Processing Application**, <https://github.com/DerWaschbar/Image-Processing>.
- [4] **Maria M. P. Petrou, Costas Petrou**, *Image Processing: The Fundamental*, 2nd edition, Wiley, 2010.
- [5] **PPM Documentation**, <http://netpbm.sourceforge.net/doc/ppm.html>.

Actor Model based Distributed Communication in Golang

Jianhao Li and Viktória Zsók

Department of Programming Languages and Compilers
Faculty of Informatics
Eötvös Loránd University
lijianhao288@hotmail.com, zsv@inf.elte.hu

The actor model [6] is inherently distributed and parallel, and it can leverage the concurrency to provide more efficiency for the distributed system [1]. The Golang programming language [5] has powerful built-in concurrency constructs such as *goroutines* and *channels*. Therefore, it is natural to think about implementing the actor model in Golang.

In this paper, we introduce a specific version of the actor model based on a practical implementation consideration.

1 Modified actor model

Now we will introduce a specific version of the actor model based on a practical implementation consideration. It follows the main philosophy of the actor model. However, some modifications are done considering the practical implementation possibility, the efficiency, and the features of the Golang programming environment.

This model aims to provide actor model-based distributed communication services for more application domains such as: common distributed software, peer-to-peer software, high-performance parallel software, and elastic cloud infrastructure.

Some new components are created in order to explain clearly the redesigned model:

1. User part and Communicator part

One actor program has its `User part` and `Communicator part`. These two parts belong to the same Golang program. The `User part` is the Golang program that needs a distributed communication service. The `Communicator part` is the distributed communication tool provided by this model. The `User part` makes decisions on the behavior of the actor. The `Communicator part` is responsible for the mail mechanisms that the distributed communication needs. The `User part` imports, configures and starts the `Communicator part` at the beginning of the program, and it uses the functions provided by the `Communicator part`, which itself will start new goroutines when it begins.

2. Mail

The `Mail` is the communication object in the usual actor model. It is a more intuitive collection of information in a specific format sent and received by actors.

3. Mail mechanism system and Mail network system.

The mail system includes two parts: the `Mail mechanism system` and the `Mail network system`. The `Mail network system` uses lower communication techniques

and provides interfaces that the `Mail mechanism system` needs.

The `Mail mechanism system` contains all the mechanisms related to the communication behavior of the actors, and it uses the interface provided by the `Mail network system`. They are decoupled so that in case of upgrading the mail network system with faster low-layer communication techniques, the whole mail system and the communication behavior of the actor system will not be affected.

4. `User mail queue` and `Communicator mail queue`.

The `User part` of an actor can have several `User mail queues`, which are the final destinations of the mails. The `Communicator mail queue` is the mail queue in the `Communicator part`. For each actor there is only one, large `Communicator queue`.

5. `Sending routing key` and `Receiving routing key`.

The `Sending routing key` is the key provided by the mail sender. The `Receiving routing key` is the routing key for each user mail queue. One actor can register different `User mail queues` with different `Receiving routing keys`. The `Mail sender` provides a `Sending routing key`. The `Communicator part` acts like the AMQP 0-9-1 direct exchange [7], it routes the `Mails` from the `Communicator mail queue` to different `User mail queues` based on the matching routing key.

6. `Mail sender` and `Mail receiver`.

The `Mail sender` is the actor sending the `Mail`, while the `Mail receiver` is the actor receiving the `Mail`.

2 New features of the redesigned actor model

In the following, the main features of this new version of the actor model are given:

Independence. The Golang compiler compiles the program into machine code. A Golang program does not need extra runtime environment like a virtual machine or an interpreter. This project does not implement a shared runtime environment for the actors in the same node, they only share the same operating system. Additionally, the actors in the same node will use different ports for communication to offer more actor independence.

Built-in group mechanisms. This model organizes actors in groups with different automatic mail forwarding mechanisms. It enables the programmers to easily form special groups with commonly needed mechanisms like group broadcasting and load balancing.

Easy to manage and monitor. This model provides management relations for actors, in case the actor agreed to be managed and it has exported some management interfaces. Whenever the manager actor sends a management mail, instead of being handled by the `user part` itself, the `communicator part` will automatically report the status, or it will be executing some other operations.

Efficiency. The mail network system is decoupled from the mail mechanism system to be able to upgrade the efficiency by supporting faster low-level communication techniques. Additionally, the mail mechanism system leverages the concurrency to provide more efficiency.

User friendly. Programmers do not need to install and run the message brokers themselves. The user only needs to import the package and to use the methods inside. This model provides more built-in mechanisms.

Now we will introduce our modifications of the classical actor model:

1. Different actor relations

The usual actor model indicates that “No actor can be operated on, looked at, taken apart or modified in any way except by sending a message to that actor requesting it perform the operation itself.” [2] In our version, the actor model has different actor relations: communication relation and management relation. The **User part** can configure the **Communicator part** to agree to be managed, to provide a management token, and to expose some management interfaces. The central management actor can manage or query the actors on different nodes by sending management messages with the token. After the **Communicator part** receives the management message, it will automatically send a status message to the management actor. The **User part** itself does not send status messages. In this model, we execute the actors in an independent, distributed way. Managing (or monitoring) the actors is done in a centralized way, in order to easily coordinate them.

2. Actor definition

The usual actor model indicates that “Every object is an actor; this includes messages and numbers.” [2] However, in this modified model every program with a mail system is an actor. The actor actually is a program, which means it can have its own behaviors. The mail system can send, receive and handle the actor communications.

3. Concurrent mail routing and handling

The usual actor model indicates that “An actor can send many messages at the same time but can only receive them one at a time. In other words the arrival order is linear; while the structure of events is only partially ordered by the notion of one event preceding another.” [2] The receiving process in this model starts when the mail arrives at the communicator mail queue and ends when it arrives at the user mail queue. Only one mail arrives at the same time to the communicator mail queue. However, after the mail arrived, the mails are routed to the user mail queue concurrently (if more CPUs are allowed to be used, then it is parallel). Therefore, in this updated model we receive the mails in parallel.

4. Organize the acquaintances differently

The usual actor model indicates that “Actor can have a set of acquaintances, other actors it knows about and can send messages to. This set can increase in time since the actor may create new acquaintances, and it may also hear about them in messages it receives. These are the only way the set of acquaintances can increase.” [2] In this model, to reduce the **User part** which needs to be implemented by the programmer, the **Communicator part** will provide more built-in mail forwarding mechanisms (the group broadcast and load balance), which are needed by distributed systems in general. To achieve that, we need to maintain more attributes of acquaintances in an **Acquaintance table** which indicates the groups each acquaintance belongs to. Additionally, we know about the type of the group by having a **Group type table**. Different types of the group have different mechanisms. According to the **Group type table**, the **Communicator part** will know how to forward the mail. According to the **acquaintance table**, the **Communicator part** knows to whom the forwarding should be done.

3 Related works

Different from the RabbitMQ [8], in this project, the queue is kept by each actor itself instead of the message broker. The routing is done by all the communicators themselves instead of the broker. Although the RabbitMQ broker supports distributed clustering, it is still not inherently distributed like in this project. It will be lighter for a peer-to-peer system to use an inherently distributed communications tool. RabbitMQ needs a plugin [9] to achieve load balance among queues on different nodes. However, in this model, the load balance among different nodes can be easily achieved by forming a group of actors with built-in load balance mechanisms.

Different from the CAF [4] and the Erlang [3], the actors in this project are more independent by letting the ones in the same node not sharing the runtime environment. In CAF and Erlang, actors leverage the concurrency and the distribution like in the usual actor model. In this model, it is the goroutines inside the actors that leverage the concurrency instead of the actors. The actor system is more focused on the efficient distributed communication among programs inside or outside of the same node with built-in communication mechanisms. In this way the programmer can form actor groups with special mechanisms (broadcasting group, load balancing, etc.), and can construct distributed systems easier.

References

- [1] **Agha, G.** *Actors: A Model of Concurrent Computation in Distributed Systems*, Tech. Rep. 844, MIT Press, USA, 1986.
- [2] **Agha, G., Mason, I. A., Smith, S. F., & Talcott, C. L.** *A foundation for actor computation*, *Journal of Functional Programming*, 7(1), 1-72, 1997.
- [3] **Armstrong, J.** *Erlang – A Survey of the Language and its Industrial Applications*, Proceedings of the symposium on industrial applications of Prolog (INAP96), Hino, pp. 16–18, 1996.
- [4] **Charousset, D., Hiesgen, R., & Schmidt, T. C.** *Revisiting actor programming in C++*, *Computer Languages, Systems & Structures*, 45, 105-131, 2016.
- [5] **Google**, The Go Programming Language, <https://golang.org/>.
- [6] **Hewitt, C., Bishop, P., & Steiger, R.** *A Universal Modular ACTOR Formalism for Artificial Intelligence*, Proceedings of the 3rd IJCAI, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1973, pp. 235–245.
- [7] **OASIS**, AMQP Working Group 0-9-1, <https://www.amqp.org/specification/0-9-1/amqp-org-download>.
- [8] **RabbitMQ**, Messaging that just works – RabbitMQ, <https://www.rabbitmq.com/>.
- [9] **Sharding Plugin**, RabbitMQ Sharding Plugin, <https://github.com/rabbitmq/rabbitmq-sharding>.

Notes on P systems versus R systems

Pramod Kumar Sethy

Doctoral School of Informatics
Faculty of Informatics, Eötvös Loránd University
Budapest, Hungary
pksethy@inf.elte.hu

Abstract

Membrane systems (P systems) and reaction systems (R systems) are two unconventional and powerful computational paradigms based on the study of the structure and functioning of living systems, in particular living cells. We first list the most important similarities and differences of the two models. Then we define a hybrid model where the components of the P system are represented by R systems and demonstrate that this new construct corresponds to particular variants of P systems. We also suggest some research directions for future investigations.

1 Introduction

The concept of a membrane system, later also called a P system, was introduced by Gh. Păun in order to develop a distributed model for computing that mimics architecture and functioning of living cells [5]. Since 2000, the idea has been extensively explored and has become rapidly developing field in bio-inspired computing (cellular computing) [6]. The main component of the generic variant of a P system is a membrane structure which consists of membranes hierarchically embedded in the outermost skin membrane. Each membrane encloses a region: a compartment, containing a multiset of objects and possibly other membranes. The objects represent bio-chemical ingredients. Each region is associated with a set of multiset-rewriting rules, to be applied to the objects in the region. The rules can be of different types, they can modify the multisets of objects in the regions and/or also can provide the possibility of transporting the objects from one region to another one. For more details about P systems readers are encouraged to consult [6].

Reaction systems, also called R systems were introduced by A. Ehrenfeucht and G. Rozenberg as a formal model of interactions between biochemical reactions; the reader is referred to [4] for the original motivation. The main idea was to model the behavior of biological systems where the functioning of a living cell consists of a huge number of individual reactions that interact with each other. The interaction between individual biochemical reactions takes place through their influence on each other, and this influence happens through the mechanisms of facilitation and inhibition. The basic model defines the possible evolution of the state of a reaction system according to a set of reactions. For further details on reaction systems we refer to [4].

Both areas deal with populations of molecules (reactants) which evolve by means of evolution rules (reactions). P systems operate with multisets of objects, while R systems work with sets of objects. In case of R systems the emphasis is always on evolution, not on computation, while in P systems' theory computation is in the center of interest. R systems

only focus on the presence or absence of the chemical species, and not on their amounts, while in case of P systems the number of the same object is significant. For this, R systems provide as a qualitative model, while P systems provide a quantitative model for the biochemical processes. Further differences between the two models are the following. In case of R systems, multiple reactions that have common reactants do not interfere. All reactions that are enabled at a certain time step are performed simultaneously. Another feature of reaction systems which makes them different from other bio-inspired computational models, as for example P systems, is the lack of permanency: the current state of the system consists of only products of those reactions that took place in the last time step. Those reactants that were not involved in any reaction disappear from the system.

One interesting question is whether hybrid models, combinations of features of P systems and R systems, provide a new model for computation with features different from features of P systems and R systems. One idea is that we replace the rule sets associated to the compartments of the P system with R systems and we allow these R systems to communicate with the neighboring compartments by sending/receiving objects, i.e. products. The concept of a communicating R system (cdcR system) with communicating products or reactions was introduced and studied in [3]. The idea of P systems working with symbol-objects without multiplicities was introduced and studied in [1].

2 Preliminaries

We recall a few elementary notions and notations that we will use in the sequel. An alphabet is a finite and nonempty set. For an alphabet V , by V^* we denote the set of all strings over V , including the empty string, denoted by λ . The set of nonempty strings over V is denoted by V^+ . \mathbb{N} is the notation for the set of natural numbers. Let O be a set of objects. A multiset is a pair $M = (V, f)$, where V is an arbitrary (not necessarily finite) set of objects from O and $f : O \rightarrow \mathbb{N}$ is a mapping which assigns to each object its multiplicity; if $a \notin V$ then $f(a) = 0$. The support of $M = (V, f)$ is the set $supp(M) = \{a \in V \mid f(a) \geq 1\}$; if $supp(M)$ is a finite set, then M is called a finite multiset. A multiset M over the finite set of objects V can be represented by any string w over the alphabet V with $|w|_a = f(a)$, $a \in V$, λ represents the empty multiset.

We first provide the notion of a standard membrane system from [6]. A P system (of degree n) is a construct, $\Pi = (O, \mu, w_1, \dots, w_n, R_1, \dots, R_n, i_{in}, i_{out})$, where O is the alphabet of objects, μ is a rooted tree, called the membrane structure (with n membranes; each node corresponds to a region), w_1, \dots, w_n , $n \geq 1$, are multisets of objects over O , w_i is the initial multiset of objects present in region i , $1 \leq i \leq n$, R_1, \dots, R_n , $n \geq 1$, are finite sets of rules associated with the regions of μ . The rules in R_i , $1 \leq i \leq n$ are of the form $u \rightarrow v$, with $u \in O^+$ and $v \in (O \times \{here, out, in\})^*$, where *here*, *out*, *in* are so-called target indications. A pair $(a, here)$ means that the object a remains in the same region, (b, out) means that b leaves to the parent region, and (c, in) means that c leaves to a child region of the considered region. Finally, i_{in}, i_{out} are the labels of input and output regions, respectively. P systems work by transitions, i.e., by changing their configurations. A configuration of a P system Π , see above, is an n -tuple $c = (u_1, \dots, u_n)$, where u_i , $1 \leq i \leq n$ is a finite multiset of objects over the set of objects O . A transition from configuration c to $c' = (u'_1, \dots, u'_n)$ means that c' is obtained from c by non-deterministic maximally parallel way of rule

applications (see [5]). A computation in Π is a finite sequence of transitions in Π starting from the initial configuration to a halting configuration, i.e., to a configuration where no rule can be applied in any of the regions. Standard P systems are very powerful computing devices, they are as powerful a Turing machines.

Now, we recall here some elementary notions about reaction systems from [4]. Let S be an alphabet and finite nonempty set; S is called the background set. A reaction(in S) is a triple $a = (R, I, P)$ where R, I, P are nonempty subsets of S such that $R \cap I = \emptyset$. R is the reactant set of a , I is the inhibitor set of a , P is the product set of a . R, I, P are also denoted by R_a, I_a, P_a . We denote by $rac(S)$ the set of all reactions in S . If $T \subseteq S$ and $a \in rac(S)$, then a is enabled by T if $R_a \subseteq T$ and $I_a \cap T = \emptyset$, then the result set of a on T , denoted by $res_a(T)$, is defined by $res_a(T) = P_a$. If a is not enabled by T , then $res_a(T) = \emptyset$. If A is finite set of reactions, then the result of A on T is defined by $res_A(T) = \bigcup_{a \in A} res_a(T)$. Then a reaction system is an ordered pair $\mathcal{A} = (S, A)$, where S is a background set and A is a finite nonempty set of reactions over S . A reaction system \mathcal{A} also operate by transitions, i.e., by changing their states. The state sequence of a reaction system \mathcal{A} with initial state T is given by successive iterations of the result function: $(res_{\mathcal{A}}^n(T))_{n \in \mathbb{N}} = (T, res_{\mathcal{A}}(T), res_{\mathcal{A}}^2(T), \dots)$.

Next we define a hybrid model when both models work together.

3 R systems in P systems - a hybrid model

In this section, we introduce a new variant of P systems, called PR systems, where we replace rules with reactions in the regions of the P system.

A PR system is a construct

$$\Pi_r = (O, \mu, w_1, \dots, w_n, R_1, \dots, R_n, i_{out}),$$

where everything remains same as in case of standard membrane system with very few exceptions. In place of rules, we consider extended reactions, i.e. reaction where the products are associated with targets. So, R_1, \dots, R_n , $1 \leq i \leq n$, are finite sets of reactions over O , R_i is associated to region i , $1 \leq i \leq n$ and each a in R_j , where $1 \leq j \leq n$ is of the form $a : (R_a, I_a, \Pi_a)$, where R_a and I_a are nonempty subsets of O , $R_a \cap I_a = \emptyset$, and $\Pi_a \subseteq P_a \times \{in, out, here\}$, P_a is a nonempty subset of O . R_a, I_a, Π_a are called the set of reactants, the set of inhibitors, and the set of products with targets.

Examining the above model, the following statement can be proven.

Theorem 1 *To any PR system a simulating P system with promoters and inhibitors can be constructed.*

The proof is based on the ideas in [2] combined with the ideas of [3]. In [2] it was shown that to any R system $\mathcal{A} = (S, A)$ where $A = \{a_i \mid 1 \leq i\}$ and $a_i = (R_{a_i}, I_{a_i}, P_{a_i})$, $1 \leq i \leq n$, a simple P system with (sets of) promoters and (sets of) inhibitors, Π , can be constructed such that Π simulates \mathcal{A} . A simple P system has only one region, the skin region; a P system with promoters and inhibitors are with rules which are associated promoters and inhibitors, i.e., objects in the presence/absence of which the rule can be applied. We construct the PR system in such way that in every regions we place rule set of a simple P

system with promoters and inhibitors and we organize the communication of the objects between the regions exactly in the same manner as it is done in communicating R systems with communication by products.

Notice that the above results can be extended to tissue-like P systems as well, i.e. P systems where the membrane structure is an arbitrary graph.

4 Conclusions

In this note, we provide a hybrid model called PR system and provided some basic statements. There are number of possible research directions for this connection between P systems and R systems. One would be to extend the model with property of membrane creation or membrane division and study the power of these variants of PR systems. PR systems where not objects but rules are communicated (like cdcR systems communicating reactions) are of interest as well.

5 Acknowledgment

This work was supported by project " Integrált kutatói utánpótlás-képzési program az informatika és számítástudomány diszciplináris területein", EFOP 3.6.3-VEKOP-16-2017-00002.

References

- [1] **Alhazov, A.**, *P systems without multiplicities of symbol-objects*, Inf. Process. Lett., **100**(3), 124 -129, 2006
- [2] **Alhazov, A., Aman, B., Freund, R., Ivanov, S.**, *Simulating R Systems by P Systems*. In: Leporati, A., Rozenberg, G., Salomaa, A., Zandron, C.,(Eds) Membrane Computing - 17th International Conference, CMC 2016, Milan, Italy, July 25-29, 2016, Revised Selected Papers, *Lecture Notes in Computer Science*, vol **10105**, Springer, Berlin, Heidelberg, 51-66, 2016
- [3] **Csuhaj-Varjú, E., Sethy, P.K.**, *Communicating Reaction Systems with Direct Communication*. In: Freund, R., Ishdorj, T.O., Rozenberg, G., Salomaa, A., Zandron, C., (Eds) Membrane Computing -21st International Conference (CMC) 2020, Vienna, Austria, September 14-18, 2020, Revised Selected Papers, *Lecture Notes in Computer Science*, vol **12687**, Springer, Berlin, Heidelberg, 17-30, 2021
- [4] **Ehrenfeucht, A., Rozenberg, G.**, *Reaction Systems*, Fundam. Informaticae, **75**, 1-4, 263–280, 2007
- [5] **Păun, G.**, *Computing with Membranes*, Journal of Computer and System Sciences, **61**, 1, 108-143, 2000
- [6] **Păun, G., Rozenberg, G., Saloma, A.**, *The Oxford Handbook of Membrane Computing*, Oxford University press, Oxford, 2010.

Error detection and analysis of PSA structured P4 programs¹

Gabriella Tóth and Máté Tejfel

Department of Programming Languages and Compilers, Eötvös Loránd University

kistoth@inf.elte.hu, matej@inf.elte.hu

In this paper, we introduce an analysis and error detection for P4 programs [10, 9], which are developed in Portable Switch Architecture (PSA) [11]. This solution is an extension of our previous work [3, 4], in which an error detection method was presented for simple P4 programs.

P4 is a domain-specific programming language to define the processing of network packets in network devices. The main information with which these programs work is the header information. P4 programs have three main parts: the *parser*, which describes the reading of the input packet and gets the header information from it; the *modifierpart*, which modifies the header information; and the *deparser*, which creates the output packet from the calculated headers. These programs have an uncommon program structure - the match-action table - which is partially defined in the source, but it is filled with specific data during runtime by an external controller, therefore the whole process can be hardly checked with static analysis approaches because there will be holes in the contents of the tables.

There are different approaches to analyze P4 programs. There are static analysis tools, which concentrate the error detection: Assert-P4 [8] checks the correctness of given conditions in annotated P4 sources by static analysis, and P4V [5] checks the satisfiability of a formula, which describes the behaviour of the P4 program. All of them works with the previous version of P4, which is the P4₁₄. There are solutions for the newer version - P4₁₆ - for example BF4 [2], which can not only detect errors in the source, but it is able to repair the code by manipulating the table contents and avoid the errors by it. There are solutions, which do not use static analysis for example P4RL [1] does runtime verification, therefore it can use runtime information. There is an approach, which is based on the dataflow analysis [6] of the program and which uses the dataflow graph to check the header usage.

Our analysis approach works only with the source code of P4₁₆ programs, and we handle the special math-action tables as branches. Our solution does not only report errors but suspicious cases too, which are cases, which may lead to error. These cases can be caused by the usage of invalid header or uninitialized fields, and it can also report the uncommon usage of packet drop. This error detection is defined for pipeline analysis, which calculates the pre-and post-condition for the different parts of the program, and analyzes these conditions.

This work is based on our previous results [3, 4], which describe an error detection of a pipeline. The main idea is to calculate a Hoare-triple, where precondition is calculated from the parser code - therefore it describes what kind of input header we would like to

¹This work has been supported by the European Union, co-financed by the European Social Fund (EFOP-3.6.3-VEKOP-16-2017-00002)

work with; the postcondition is calculated from the source of deparser - so it describes what kind of output header information we would like to forward to the network; and from the modifier part we calculate the main program. Based on this triple, we can check if our program starts from any initial state, it can reach one of the final states. If there is an execution path, which can not reach then we can find a case to report. In this paper, we would like to extend this idea to PSA structured programs too.

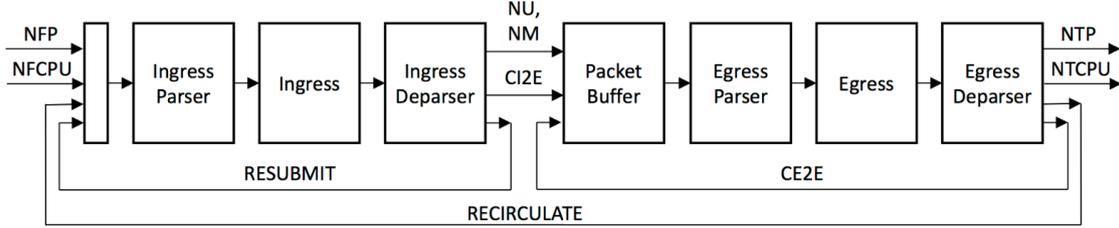


Figure 1: Packet processing paths [11]

The model of PSA can be seen in Figure 1. In PSA structured programs, there are two pipelines, the ingress, and egress pipeline, which can be analyzed by our method separately. In the figure, we can see the possible packet processing paths, we work with four of them: CI2E, Resubmit, CE2E and Recirculate. All of them sign a connection point between the pipelines, and they have a source and a destination. CI2E is the "clone from ingress to egress", its source is the ingress deparser and destination is the egress parser. The source of Resubmit is the ingress deparser and its destination is the ingress parser. C2E2 is the "clone from egress to egress", where the source is the egress deparser and destination is the egress parser. The source of the Recirculate is the egress parser, and the destination of it is the ingress parser. We would like to check if these connections are right, and we use the pre-and post-condition of the ingress and egress pipeline for it. These conditions are calculated in the pipeline checking and we use them to check a consequence for every connection point. Every consequence checks if the condition of the destination is a consequence of the source, which is right if the condition of the source is true then the condition of the destination is also true. To check the consequence, we can change the expression of every consequence to a formula, where the consequence formula is negated and that is conjugated with the source formula. This formula is unsatisfiable if and only if the consequence is right. These expressions can be seen in Figure 2, where $FormulaSet \models Formula$ means the semantic consequence and the meaning of the $Pre_{I/E}$ and $Post_{I/E}$ are the pre- and the post-condition of the ingress/egress pipeline.

Path	Consequence	Formula
CI2E	$\{Post_I\} \models Pre_E$	$Post_I \wedge \neg Pre_E$
Resubmit	$\{Post_I\} \models Pre_I$	$Post_I \wedge \neg Pre_I$
CE2E	$\{Post_E\} \models Pre_E$	$Post_E \wedge \neg Pre_E$
Recirculate	$\{Post_E\} \models Pre_I$	$Post_E \wedge \neg Pre_I$

Figure 2: Consequence and formula for path checking

An example can be seen in Figure 3, where the conditions only contain the header name, which should be valid. A real pre- and post-condition have a similar structure. The example

contains three types of header: *ethernet*, *ipv4* and *myTunnel*. The figure only shows the formula, that should be checked to be unsatisfiable. In the case of CI2E and CE2E, it can be easily seen the unsatisfiability is coming from the *ethernet* header, because in every case of the source condition it should be valid, and the destination condition needs the same, so we see it in a negated form, therefore both formula will be false in every possible interpretation. In the case of Resubmit, the condition of the source contains two execution paths. Both of them have a valid header *ethernet*, one of them has a valid *ipv4*, other has a valid *myTunnel*. The condition of the destination needs an *ethernet* and *ipv4*, which should be negated in the formula. We can see, in the first execution path of the source, the formula could be unsatisfiable, but in the second execution path, there is a satisfiable case, when *ethernet* and *myTunnel* is valid, but *ipv4* is invalid. In this interpretation the consequence is not right, therefore we could report it to be rechecked by the developer.

$$\begin{aligned}
 Pre_I &= ethernet \wedge ipv4 \\
 Post_I &= (ethernet \wedge ipv4) \vee (ethernet \wedge myTunnel) \\
 Pre_E &= ethernet \\
 Post_E &= ethernet \wedge ipv4 \wedge myTunnel \\
 \\
 CI2E: & \quad ((ethernet \wedge ipv4) \vee (ethernet \wedge myTunnel)) \wedge \neg ethernet \\
 Resubmit: & \quad ((ethernet \wedge ipv4) \vee (ethernet \wedge myTunnel)) \wedge \neg(ethernet \wedge ipv4) \\
 CE2E: & \quad ethernet \wedge ipv4 \wedge myTunnel \wedge \neg ethernet \\
 Recirculate: & \quad ethernet \wedge ipv4 \wedge myTunnel \wedge \neg(ethernet \wedge ipv4)
 \end{aligned}$$

Figure 3: Example path checking

This idea is being integrated into the P4Query [12], which is an analysis framework for P4 programs, which is developed in a project of ELTE. Our goal with this program is to make a useful tool for P4 developers from which they can get reports about their program and its correctness. It has already contained the error checking for one pipeline, and it can work with PSA structured programs too. For the unsatisfiability checking, we use the Z3 Theorem Prover [7]. If the solver gives an interpretation, where the formula is true, it means it is satisfiable, therefore there can be an execution path, where the consequence will not be true. We will report these kinds of suspicious cases, with the model data, which describes the possible execution path, so the developer can fix the source if this is a problem.

Another supplementation of our previous work is to check the usage of the drop function, which means we would like to drop the packet. The function of the drop only set the *outputport* metadata into a *drop_port* value, which means we drop the packet, but during the further runtime, it can set it to be not dropped. We would like to report this behaviour too, and those cases, when more drop is used, while there is not any reset, because these can be suspicious, so we would like to draw the attention to them, and the developer can decide if they are correct or not.

Our pipeline checking can be extended by this checking too, by watching the *drop* predicate in the conditions, and when we see a drop function, while the *drop* condition is true, then we can see a multiple dropping. We can check if the port is changed after the drop, and we can report that the drop was reset.

With these extensions, we would like to expand the possible error checking of P4

programs, using static analysis, while only using the raw sources. We plan to refine the checking in PSA programs, to be able to give a more proper report about it and a further plan is to give refactoring ideas for the developer, and automatic solutions to reduce the work after the report.

References

- [1] **Apoorv Shukla, Kevin Nico Hudemann, Artur Hecker and Stefan Schmid** *Runtime Verification of P4 Switches with Reinforcement Learning*, ACM, New York, 2019.
- [2] **Dragos Dumitrescu, Radu Stoenescu, Lorina Negreanu and Costin Raiciu** *Bf4: Towards Bug-Free P4 Programs*, Association for Computing Machinery, New York, 2020.
- [3] **Gabriella Tóth and Máté Tejfel** *A formal method to detect possible P4 specific errors*, PTI, Germany, 2019.
- [4] **Gabriella Tóth and Máté Tejfel** *Component-based error detection of P4 programs*, [accepted but not published], Acta Cybernetica, Szeged, 2021.
- [5] **Jed Liu, William Hallahan, Cole Schlesinger, Milad Sharif, Jeongkeun Lee, Robert Soulé, Han Wang, Călin Cașcaval, Nick McKeown and Nate Foster** *P4V: Practical Verification for Programmable Data Planes*, ACM, New York, 2018.
- [6] **K. Birnfeld, D. C. da Silva, W. Cordeiro and B. B. N. de França** *P4 Switch Code Data Flow Analysis: Towards Stronger Verification of Forwarding Plane Software*, NOMS 2020, IEEE, 2020.
- [7] **Leonardo De Moura and Nikolaj Bjørner** *Z3: An Efficient SMT Solver*, Springer-Verlag, Berlin, 2008.
- [8] **Lucas Freire, Miguel Neves, Lucas Leal, Kirill Levchenko, Alberto Schaeffer-Filho and Marinho Barcellos** *Uncovering Bugs in P4 Programs with Assertion-based Verification*, ACM, New York, 2018.
- [9] **Mihai Budiu and Chris Dodd** *The P4₁₆ Programming Language*, ACM, New York, 2017.
- [10] **Pat Bosshart, Dan Daly, Glen Gibb, Martin Izzard, Nick McKeown, Jennifer Rexford, Cole Schlesinger, Dan Talayco, Amin Vahdat, George Varghese and David Walker** *P4: Programming Protocol-independent Packet Processors*, ACM, New York, 2014.
- [11] **P4₁₆ Portable Switch Architecture (PSA)**, <https://p4.org/p4-spec/docs/PSA.html>, 2020
- [12] **The GitHub repository of P4Query** <https://github.com/P4ELTE/P4Query>, 2020.

Author Index

- Ágoston, P., 65
Benczúr, A., 11
Bakos, B., 23
Bayleyegn, T., 165
Biró, Sz., 183
Csanády, B., 187
Boda, L., 169
Bozó, I., 251
Csiszárík A., 199
Csomós, P., 139
Damásdi, G., 69, 73
Donkó, I., 213
El Khalfaoui, S., 41
Erdei, Zs., 251
Faragó, I., 177
Fekete, I., 151
Frankl, N., 69
Garamvölgyi, D., 87
Grolmusz, V., 241
Grdzlishvili, B., 255
Gyarmati, M., 37
Hadjimichael, Y., 177
Haffner, D., 143
Havasi, Á., 165
Hegyvári, N., 23, 27
Héger, T., 47
Hidy, G., 147, 191
Horváth, B., 111
Horváth, R., 161, 177
Izsák, F., 135,139,143
Jackson, B., 83
Jordán, T., 91
Kaposi, A., 213, 217
Karl, J., 61
Kaszanitzky, V. E., 83
Károlyi, G., 195
Keresztes, L., 233
Kherbouche, M. 115
Király, Cs., 95, 99
Kiss, M., 199
Kiss, R., 51
Kovács, A., 223
Kraus, N., 209
Levene, R., 77
Li, J., 259
Ligeti, P., 37
Lukács, A., 187, 191, 195
Maga, B., 199
Matszangosz, Á., 199
Mihálykó, A., 95, 99
Molnár, A., 151
Molnár, B., 105, 111, 115,125, 125,129
Mukashaty, A., 115
Nagy, G. P., 41, 51
Nagy, Z. L., 47
Pach, P. P., 19
Palincza, R., 31
Pálffy, M., 23
Pálvölgyi, D., 73
Perczel, A., 231
Pituk, S., 55
Schulze, B., 83
Seifu, B., 129
Sethy, P. K., 263
Simon, P., 151
Svantnerné Sebestyén, G., 173
Sziklai, P., 37
Szögi, E., 237
Takács, B., 177
Takács, K., 241
Takáts, M., 37
Tejfel, M., 267
Tóth, Géza, 61
Tóth, Gabriella, 267
Tóth, M., 251
Varadarajan, N., 77
Varga, B., 245
Varga, D., 199
Varró, Sz., 183
Yinghong, Z. 125
Zolotareva, E., 129
Zongpu, X., 217
Zsók, V., 255, 259
