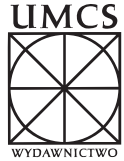
vol. III

# Selected Topics in Applied Computer Science

EDITED BY Jarosław Bylina
Aneta Wróblewska

Maria Curie-Skłodowska University Press

Selected Topics in
**Applied Computer Science**


UMCS
WYDAWNICTWO

vol. III

# Selected Topics in Applied Computer Science

EDITED BY Jarosław Bylina
Aneta Wróblewska

Maria Curie-Skłodowska University Press
Lublin 2024

Reviewers
*Mikołaj Deckert*
*Katarzyna Filus*
*Dariusz Mikołajewski*

Cover and front page design
*Krzysztof Trojnar*

Typesetting
*Jarosław Bylina*
*Aneta Wróblewska*

Cover illustration: pixabay.com, freepik.com



ISBN 978-83-227-9872-0

Maria Curie-Skłodowska University Press
ul. Idziego Radziszewskiego 11, 20-031 Lublin, Poland
tel. +48 81 537 53 04
www.wydawnictwo.umcs.eu
e-mail: sekretariat@wydawnictwo.umcs.lublin.pl

Sales Department
tel./fax +48 81 537 53 02
Online bookstore: www.wydawnictwo.umcs.eu
e-mail: wydawnictwo@umcs.eu

# Contents

# Preface

The creation of this book was inspired by the growing complexity and interdisciplinary nature of modern computational research, spanning fields such as linguistics, neuroscience, artificial intelligence, and epidemic modeling. In a world where technology continues to integrate itself into our lives and societies, understanding its foundations and practical applications has never been more critical. This volume brings together contributions from diverse research domains, showcasing innovative methods, tools, and theoretical advancements that address contemporary challenges in computation and its applications.

One of the key motivations for this collection was to bridge the gap between theory and practice. Each chapter provides not only theoretical insights but also delves into the application of these ideas in real-world contexts. For instance, the development of Picto, a software system for studying language emergence, demonstrates how experimental linguistics can benefit from computational tools to explore communication dynamics. Similarly, contributions focused on EEG signal analysis illustrate how machine learning can enhance our understanding of the human brain and improve applications in medicine and psychology.

The interdisciplinary nature of this work reflects a broader trend in academia and industry: the need for collaboration across traditional disciplinary boundaries. By bringing together experts in sociolinguistics, algorithmic efficiency, neural networks, and epidemic modeling, this book exemplifies how diverse perspectives can converge to solve complex problems. The chapters are carefully curated to offer both depth in individual topics and a holistic view of the interconnectedness of these fields.

Another significant theme in this book is the emphasis on fostering autonomy and creativity in solving problems. Whether it is through exploring the autonomy-driven approach in teaching software localization or addressing the nuances of language emergence in children, the contributors highlight the importance of designing flexible, user-centered methodologies. These approaches not only advance academic understanding but also pave the way for practical innovations that can be readily implemented in education, healthcare, and technology development.

We would like to express our gratitude to the authors for their contributions and to the reviewers whose feedback enriched the quality of this book. This volume is the result of collective efforts, reflecting the shared commitment to advancing knowledge and solving pressing challenges through research. It is our hope that this book will serve as a valuable resource for researchers, educators, and practitioners alike, sparking new ideas and collaborations.

Finally, it is impossible not to thank the many other people who were involved in the creation of the book — namely, everyone who contributed to its publication from the editorial and technical side.

We believe this book will capture the interest of anyone intrigued by the continuous evolution of computer science, particularly students and researchers eager to explore its myriad applications.

*Jarosław Bylina*
jaroslaw.bylina@umcs.pl

# Developing Picto: Comprehensive Language Emergence Research Software

Jan Bylina
Piotr Kosela*
Oskar Maksim
Mikołaj Martyna
Katarzyna Woźniak

## 1 Introduction

The field of sociolinguistics, researching the influence of social structures on language development, has garnered high levels of interest, but remains insufficiently explored. There are numerous experimental semiotics studies on adults [11], but the lack of sufficient data on children and adolescents prevents us from performing reliable differential analysis of the two groups. Data concerning humans at the age of peak language acquisition capabilities, in with accordance the Critical Period Hypothesis [5, 13], could provide valuable insights into language emergence. Efficient data collection and simplified post-processing and interpretation are integral components of undertaking this type of research. Therefore, a system that can facilitate these tasks is crucial. The purpose of this paper is to present the design and implementation of Picto: a system designed to help future studies in selecting the most appropriate solutions.

Removing the influence of the natural language is one of the main concerns in experimental language development research. The way of solving this problem greatly affects the experiment's design. Different solutions impose different limitations and change the focus of the research. When designing an experiment, this decision needs to be taken with great precaution.

The main inspiration for our project's architecture is celebrated [3]. In this research, pairs of participants were playing a cooperative game, requiring communication. However, players could not use their natural language. They were forced to invent a novel way of communication, using a visual medium, such that no alphabetical or iconical signs could be used. Participants were writing on the pads

*Corresponding author — piotr.kosela@mail.umcs.pl

and their partners were receiving distorted live pictures of the pad. Like in natural language, signals in Galantucci's setting quickly faded and provided many dimensions for coding: not only shape but also timing, thickness, and location on the panel. However, we were about to minimize signals' dimensionality, so large groups of children could faster converge to communicational consensus.

[1] study influence of feedback and imitation on communication system emergence. Participants were matched in pairs. One member of each pair had to draw an image depicting some, often abstract, meaning. The second participant guessed the message from the provided list of answers and with the given drawing. This research focuses on sign complexity. Our concern with this way of realizing communication is that young participants of our experiment could on purpose draw misleading images. So we have decided to provide players only with a pre-designed set of abstract images.

Another medium, which can be used to produce artificial communication is sound. [6] provide participants with two buzzers, producing different notes. Sequences of the sounds are used to signal meaning, which is one of the abstract, colorful shapes presented. Both adults and children are taking part in the study. Researchers report better performance of the former when it comes to learning and using buzzer language. However, memorizing, up to 7 sounds long, binary sequences is cognitively demanding. Adults' long-term (over 2 minutes) memory works much better than children's [2]. Thus, we have designed our medium to minimize the effort put into memory, providing participants with as many aids as possible.

Due to the short attention span of children participating in our study, and to easier engage them, we have aimed to make the game as easy, as possible, while staying with a reasonable challenge. According to [9, 10] shared context positively affects the development of communication. Note that Müller's color framework closely resembles our system, but we significantly reduce the number of possible signs and presented stimuli.

One of our long-term aims is to examine the influence of social structures — represented by the frequency of interactions — in the development of language. Discovering and description of this relation in field research may be traced back to, at least, [8]. It was extensively studied using computational models, e.g. [15, 4]. Applying such a paradigm when experimenting with humans, especially children, leads to complications. Fatigue and short attention span seriously limit the number of games a group can play in one session. Such trials were reported on in [12]. Picto is meant to be a framework for similar experiments, but appropriate for large groups of children. This leads to the problem with waiting time. When partners to consecutive games can be chosen freely, one can match participants with any other player available. It is not the case when frequencies of interactions between specific people need to obey some given pattern. Minimizing waiting time and keeping children's attention was the most challenging part of Picto development.

To justify some of our design decisions, we start the paper with a short review of the subject's literature. Then we describe the course of the experiment from the participant's perspective. According to that, we identify a list of features desirable for our software. Next, we proceed to the detailed description of Picto's implementation and present sample results collected using our system. Finally, we summarize with conclusions regarding the technical details of the software.

## 2 Course of the experiment

In our setting, the experiment takes place in two separate computer rooms, and all participants are randomly divided into two groups. Due to study participants being native Polish speakers, all Picto communicates are written in this language, as one can see in Figure 1. Before the experiment's start, all participants have undergone short training familiarizing with Picto's interface.

Each participant takes a seat at their computer. On each of the workstations, there is the starting screen of the application prepared. Two games are run parallelly, mixing participants from both rooms. The facilitators check if all participants are ready and remotely initiate the experiment. The game begins. The players are paired, in compliance with the prepared social structure.

Each pair consists of a speaker and a listener. On the speaker's screen, Figure 1a, there are four images, one of them highlighted, and four columns with two symbols each. At the same time, the listener's screen displays a board with the message 'waiting for the speaker' and a timer, presented in Figure 1c. The speaker's task is to communicate the highlighted image using the symbols given. From each column, the speaker must choose one symbol. On the right side symbols are displayed overlaid, creating a 'word', which, according to the speaker, describes the highlighted image best. Figure 2 presents an example word created by the speaker. The speaker confirms it by clicking the 'send' button. On the listener's screen, the final 'word' created by the speaker is displayed, along with the four images again. During this time, the speaker's screen shows a board with the message 'waiting for the listener' and a timer, analogous to the one depicted in Figure 1c.

The listener's task is to choose one of the displayed images, which, in their opinion, is represented by the given word. The chosen image is highlighted, and the 'send' button appears. Then, on both the speaker's and listener's screens, a board with feedback on the correctness of the listener's response appears, indicating 'correct' or 'incorrect', Figures 1e and 1f. After this, participants move on to new interlocutors, and the whole process repeats.

## 3 Objectives

The Picto system was designed to provide a framework for collecting data about the social dynamics of language learning in an experimental setting. For this purpose, before developing the system, we established a clear set of objectives and goals that the system must fulfill to guarantee the accurate execution of our primary research. We aimed to create a complete, self-contained platform that was tailored to our needs, highly customizable and allowed us to change experiment parameters on the fly between research groups.

The main issues, primarily because of the volume and complexity of the data, were the need for robust data collection protocols and a reliable database system, both of which required careful consideration. Our study also required the system to engage participants effectively without causing a loss of focus. While this has not presented a challenge in adult studies, managing the attention of the young population has proved difficult at times and required careful consideration.

The primary aim of the Picto tool is the systematic recording of every aspect of artificial language emergence. The objective of the system is to furnish data on the participants' responses, messages they produce in response to specific stimuli, their response time, and to gauge the general dynamics within their distinct, regulated population.

Data collection was from the very beginning an essential aspect of Picto, with the primary objective of accurately reconstructing the game's progression based solely on the gathered data. Numerous parameters and variables, such as symbol selection, images and response times, were carefully evaluated to achieve the set objective, and rigorous testing was carried out to ensure the expected stability and quality of the data. Recorded information enables us to recreate any chosen moment of each game played.

Picto was designed to be a self-contained platform that could be easily deployed and managed. This required the creation of a stable infrastructure that could be easily maintained and managed by a single person. The system was designed to be highly customizable, allowing for the modification of parameters on the fly between test groups. This was achieved by creating a robust database system that could be easily accessed and modified by the administrator.

The research was carried out in a rapidly changing environment with a large user base, consisting of children aged from 9 to 16, who had limited time to familiarize themselves with and adapt to our platform. Consequently, we required an intuitive and user-friendly interface that could help us meet all of our research goals. This necessitated prioritizing the familiarity principle of graphic and UI/UX design above others.

Maintaining the attention of our participants was a crucial objective, particularly given their young age. In our case, this proved to be challenging. The median age during the test phase was $M = 11$, while the mean and standard deviation were $\mu = 11.94$ and $\sigma = 2.02$, respectively. This created the need to carefully examine our methodology and approach, which arguably led to the improvement of the quality of the study itself. Low waiting time and interface simplicity became crucial to achieving this goal.

# 4 General approach

In this section, we describe the Picto infrastructure. We briefly describe our working environment and justify choose of specific solutions. Then we take on a detailed analysis of two main components of our system: frontend and backend.

To ensure the stability of the system, we have decided to use a dedicated virtual machine with a static IP address. The server was configured to run Docker containers (Spring App, Nginx, PostgreSQL) and was equipped with a 100GB hard drive and 8GB of RAM. The machine was equipped with an Uninterruptible Power Supply (UPS) to protect the system from power outages.

To provision the server, we have used Ansible, a configuration management tool that allows for the automation of the deployment process. Ansible was used to install all the necessary packages and to configure the server. The configuration files were stored in a Git repository, which allowed us to easily track changes and revert to previous versions if necessary. This also allowed us to easily deploy the

system onto a new server if the need arose. We have used that feature to provision development and testing servers.

Ansible was integrated with Github Actions, a CI/CD tool that enabled automatic system deployment on the server following each commit. This enabled us to test the system on a development server and to swiftly deploy it on the production server once the tests had passed.

## 4.1 Technological stack

We have had multiple choices for a front-end library to use. We have considered React, Angular and VueJS. While all of these frameworks are designed to build single-page applications, we have decided that React is the most flexible choice. It is also developed by Meta and used by many leading technology brands, so it is reliable and there is plenty of learning material on the internet. It also offers cross-platform support in case we would like to port our application.

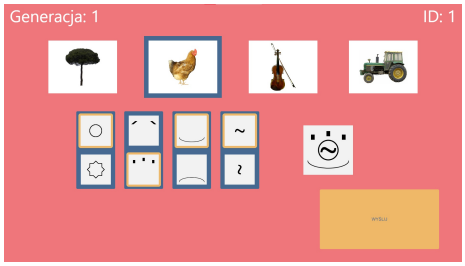
Selecting the optimal technology stack for our backend proved challenging. Various solutions and frameworks were created for REST API services, making the decision more complex. We have evaluated several options: Python with Django, Flask, or FastAPI; Node.js with Express.js; PHP with Laravel; Ruby with Ruby on Rails; and Java with the Spring Boot framework. After careful consideration of all of these languages and frameworks, and comparing them to our goals, we have concluded that the latter would be the best fit for us. This was due to several reasons. The key factors were the ability to write easily readable, sustainable, object-oriented code with minimal boilerplate (Lombok was utilized to decrease the number of lines required for constructors and builder classes), stability to handle heavy workloads, and the comprehensive support network.

Choosing the appropriate database management system was based on its ease of use and ability to handle large volumes of data, with data consistency and normalization being important considerations. It was concluded that an SQL database would be preferable to the NoSQL counterpart [7], despite a potential compromise in performance. This tradeoff was deemed reasonable to ensure data consistency, integrity, and compliance with the ACID model.

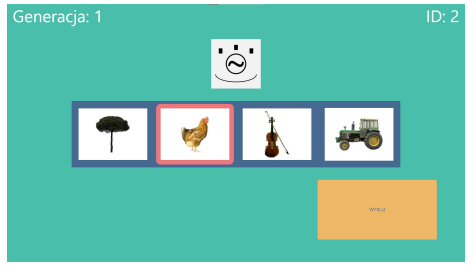
## 4.2 Frontend

Frontend needs to be intuitive and attract attention. To achieve this objective, we have applied a user-centered design approach, using standard UI/UX and graphic design principles, including visual hierarchy to emphasize important elements with strong contrast. We have also maintained consistency between the speaker and listener sections and opted for minimalism in adding only essential experiment elements without embellishment. During the test phase, we implemented a timer to display user status and waiting time.

The implementation is based on the React framework, chosen for reusable and flexible components. It allows the site to have much less code and be more dynamic. Additionally, we have used a library called Material UI for its abundance of a lot of ready-to-use components and popularity, achieving familiarity for users through uniformity with other websites. To modify the front end's style, we have




(a) Speaker's screen




(b) Listener's screen


(c) Waiting timer


(d) Finish screen


(e) Correct screen


(f) Wrong screen

Figure 1: Screenshots of Picto's interface (all text is written in Polish, which was the native language of the participants)

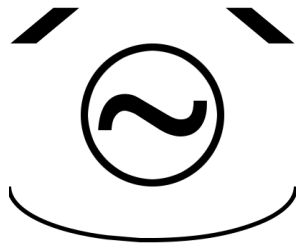


Figure 2: An example of a word created by speakers to describe a given image — this word was the most frequently produced by participants across all trials

created CSS classes and assigned them to components conditionally. For example, all components of the `imageSelected` class are to be given a yellow border of one view width. All the components we have implemented are JavaScript Functional Components because they are much simpler than Class Components. Like every function, they can have their arguments and a return value.

The main component of the implementation is the `User` component. It stores all the needed states like `userId`, `generation` or the current `userState`. It is also responsible for event handling, because of its access to all the important data. The return value of the `User` component is another component, depending on the current game state. For example, if the current state is *'join'* then a `joinUser` the component would get returned.

The main states are: *'speaker'*, *'listener'*, *'join'*, *'result'*, *'waiting'* and *'end'*. All of them except *'end'* have the `userId` and `generation` displayed at the top. The *'speaker'* state, associated with the `speakerComponent`, displays images to be described, groups of symbols to be selected, a white square with all the symbols superimposed on one another and the 'send' button, which is shown only if all the required symbols are selected, so that an empty symbol list cannot be sent. The *'listener'* state is very similar to the speaker, with the caveat of not displaying a symbol selection component, instead having a selectable list of images at the top. The *'join'* component is a big 'join' button with the input field for a game ID. It is mainly seen by researchers, as they set the experiment up. Instead of joining the game immediately, however, there is an intermediary step. The user needs to check the box, acknowledging voluntary participation and giving consent to record the course of the game. This is the moment when a join call is sent to the backend. The *'result'* component comprises a large text that displays 'CORRECT' or 'WRONG', based on the image chosen by the listener. The *'waiting'* and *'end'* states are just to inform the user if they are waiting, or have already finished the game.

To achieve reusability and simplify debugging, we have implemented a handful of generic components. We have used the `PictureComponent` and `PictureListComponent` for any picture, symbol or stimuli, that needs to be displayed. On the other hand, we have utilized the `PictureToggleButtons` component for picture groups where only one of the pictures could be selected, be it symbols for speakers, or images for listeners. The `AllSelectedSymbolsComponent` superimposes the symbols, that had their IDs passed in the `chosenSymbols` argument or all of the symbols in the `selectionSymbols` argument if the `selectAll` flag is up. The `InfoComponent` is responsible for displaying the user ID and generation for every appropriate screen. The admin panel contains a lot of `ElementConfigComponent`, which all comprises a name and an input field, and `CheckBoxConfigComponent`, which works similarly but sets boolean values.

Most of the API's communication layer was implemented using the Axios library. It offers an easy way to ensure that all the communication will be made with the same backend, without repeating parts of code. Another element we have used is an EventSource. We have used it as an event listener for events sent from the backend. We also need to pass a token as a query parameter to authenticate. We use most of the data acquired from the backend 'as is', without changes. The only exception is the `setSymbolsFromBackend` function, which needs to map IDs coming from the backend, to IDs used by the frontend.

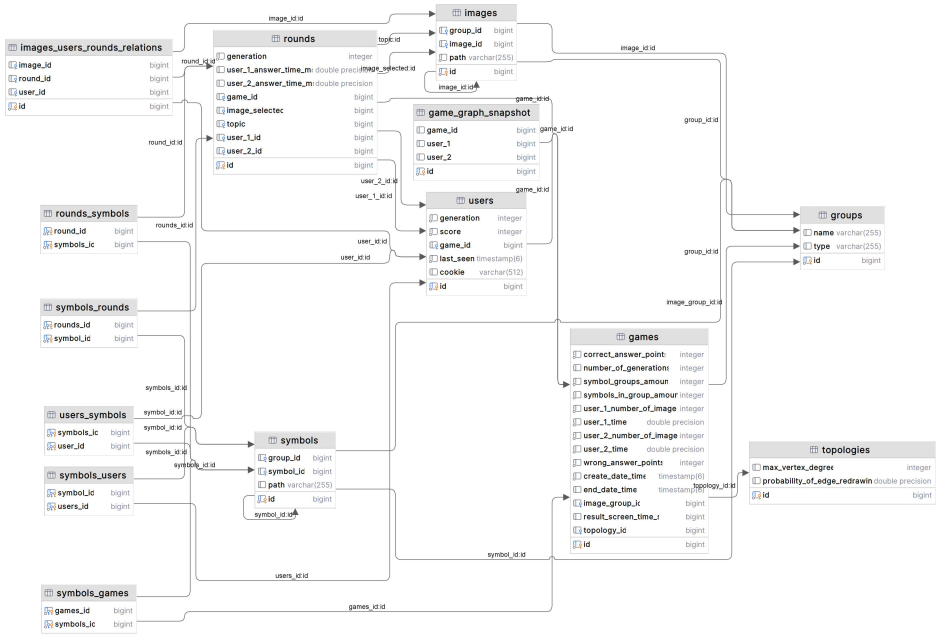



Figure 3: The schema of Picto's database

## 4.3 Backend

The backend serves as the core of our application with vital responsibilities, including managing all communication from the front end, ensuring data security and integrity, and enabling the games to be carried out. To achieve these goals, while enhancing code readability and future maintenance, the backend design is structured into three primary components: API, Service, and Repository layers. Each layer is assigned to the specific functions, critical for the system to operate seamlessly. Strict adherence to the SOLID design pattern was vital in creating a maintainable and easy-to-manage application, while still being able to implement new features as needed.

To supply enough data for future study and facilitate changes during the experiment, the database had to store every possible detail of the game flexibly. We have not used some of this flexibility yet, like for example the ability to include an arbitrary number of symbols and symbol groups, but it holds the potential for aiding in future experiments. The database was designed to meet the requirements of the third normal form; its schema is depicted in Figure 3.

The API layer acts as a gateway to the backend and, consequently, the database. It enables front-end to query for needed data, while also ensuring the data's integrity and strict adherence to specified input and return types. Picto uses REST controller from Spring Framework to implement this layer, with each endpoint having HTTP `POST`, `GET`, `PUT`, and `DELETE` requests specifically assigned to it, ensuring that only the necessary methods are utilized. The endpoints are divided into five main groups, each with its tasks and responsibilities.

Endpoints are grouped into five paths: `/game/`, `/round/`, `/user/`, `/image/` and `/symbol/`. These paths all originate from the root and each has its specific responsibilities. For example, the `/game/admin/create` path calls the method for creating a game and utilizes `admin/` for access control, specifically to prevent non-authorized parties from modifying game states. One exception to this principle is the `/event` endpoint, used for server-sent events, which does not require the consolidation of any functionalities and is, therefore, self-contained.

An implementation of the `/game/admin/create` endpoint is provided below. It should be noted, that the GameController class, which contains the controller, is annotated with `@RequestMapping("game/")`, therefore the path below is relative, not absolute.

```
 @PostMapping("admin/create") public @ResponseBody Game
createGame(@RequestBody Game game) {
    return gameService.createGame(game);
}
```

The service layer is responsible for providing functionality to all endpoints and handling the data provided. Each set of endpoints has its service class that aggregates all the functionality associated with a given part of the system and enables efficient and structured communication with the repository layer. For example, the `createGame()` method introduced earlier uses the `gameService` object to communicate with the repository and perform all the business logic required to create the game. The return value is the `Game` object that is saved to the database by the respective repository. Such a template is implemented throughout the system, making it consistent and easy to maintain.

The repository layer provides an additional level of abstraction over data access. It allows the underlying database system to be modified during development, without having to rewrite queries that have already been created. Hibernate, the implementation of the JPA repository we have used uses objects to represent entities, which are used for object-relational mappings that convert the object into a database table, the object's fields into columns, and fields with `@OneToOne`, `@OneToMany`, `@ManyToOne`, and `@ManyToMany` annotations into relations.

An example of such an object in the Picto system, specifically the `Topology` entity, is presented below. The `@Id` and `@GeneratedValue` annotations are applied to create a distinct identifier for each record, while the `@OneToMany` annotation indicates a one-to-many relationship with the `Game` entity. This is because one topology can be assigned to various games, but each game can only have one corresponding topology. The `@Column` annotations specify the fields to be converted into columns and specify their corresponding names within the database.

```
@Entity @Table(name = "topologies") public class Topology {
   @Id @GeneratedValue(strategy = GenerationType.IDENTITY)
       private Long id;

   @Column(name = "probability_of_edge_redrawing") private double
   probabilityOfEdgeRedrawing;
```

```
    @Column(name = "max_vertex_degree") private int maxVertexDegree;

// Relations @OneToMany(mappedBy = "topology",
        cascade = CascadeType.ALL)
private Set<Game> games; }
```

The entities operate as standard objects but offer the extra advantage of being transactional with the repository layer. One crucial difference is that the topology can be saved in the database through the repository using the `topologyRepository.save(topology)` method, where the `topology` object is an instance of the `Topology` entity class.

The repository layer is also responsible for automatically managing transactions. This involves overseeing the commits to the database to optimize the load and enhance the overall performance of the database system. This is generally highly desired and helpful, as it reduces the overall strain on the system. However, this mechanism has proven to be quite challenging to overcome in certain specific circumstances. In our scenario, we have required the ability to rapidly, almost simultaneously, store and retrieve the previously saved data from the database. Due to automatic transaction management, and the lack of override options, we were forced to synchronize object states without depending solely on the database for accurate states at all times.

The issue arose when attempting to retrieve data about the user's generation from the database. Our `User` object stores information about what generation the user is on. Initially, after the frontend API communication layer submitted a `GET` request to the API on the `/round/next/{userId}` endpoint, the service layer queried the database for a user with the specified `userId`. It extracted information on the user's current generation from the `generation` field. The system assumed the information to be true and waited for the speaker and listener to align, which was sometimes unattainable. This was occurring because the speaker and listener were already aligned, but the data was not retrieved due to the slow nature of data commitment by the repository.

The issue occurred when both parties attempted to query for each other's generations simultaneously. Due to the automatic transaction mechanism within the JPA Repository, there was not enough time to commit all the data, including the new generation of each user. Consequently, both parties ended up waiting indefinitely. To resolve this problem, due to the absence of a better mechanism, we have created a `userGenerations` HashMap, that stores all user generations. The benefit of this approach, as opposed to retrieving data from the database, is the assurance of up-to-date and reliable data. This straightforward solution facilitated a complete resolution of the issue.

We have decided to implement a social network using, so-called, small world graphs [14]. These are proven to effectively simulate interaction frequencies of real-world social groups [15]. Let $n \in \mathbb{N}$ denote number of participants, $k \in \mathbb{N}$, $p \in (0,1) \subset \mathbb{R}$ are parameters. We obtain small world graph $\Gamma(n,k,p)$ following this procedure:

1. Start with $k$-regular, simple graph on $n$ vertices; let $E$ denote its edge set.

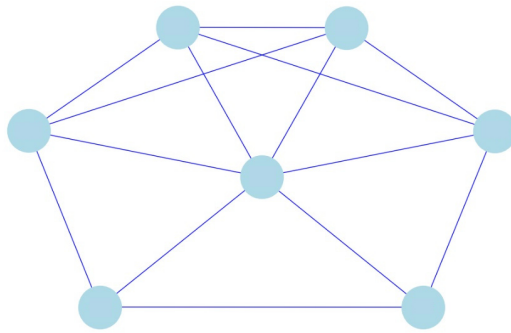


Figure 4: Small world graph representing sample social topology used in the study; generated with $k = 4$ and $p = 0.12$

2. For each edge in $E$, with probability $p$, change exactly one of its ends to a different, chosen with uniform probability, vertex.

Figure 4 presents an example of a small-world graph generated by this algorithm.

Using such a graph to determine the pairing frequencies of people may become problematic. In the case of computer simulation, one could choose any edge at random, perform interaction with its ends and repeat. When working with people, to keep them engaged, one needs to obtain as many separate pairs as possible so that every participant can play the game simultaneously; assignments also need to differ from generation to generation. As far as we are aware, there is no deterministic algorithm calculating such a subset of $E$. Our solution is straightforward: we take random, with uniform probability, edge $(a, b) \in E$. Users $a$ and $b$ are going to be matched together. Therefore, we discard all edges $(u, v)$, such that $u \in \{a, b\} \vee v \in \{a, b\}$. We repeat this random edge choosing until $E \neq \emptyset$. Then we randomly match any participants left, with uniform probability.

# 5 Results

Here we provide some results obtained using the Picto system. The main measure calculated in the study is Communicative Success (CS), which is a ratio of successful interactions number to number of all interactions in a given generation. Figures 5 and 6 present temporal change of that value, with single generation as a unit of time.

The data was collected from different research groups, with slightly different Picto settings, although both were operating on topology generated with $k = 8$ and $p = 0.12$. The first group, depicted in Figure 5, consists of 29 participants, all 16 years old. These participants started with a rich set of meanings to communicate, which was changed to a smaller, more differentiated, one after the break. Participants, whose results are depicted in Figure 6, were 20 children, all 12 years old. They started with a small set of stimuli and were moved to the bigger set.

One can easily notice, looking at the regression line, the difference between the

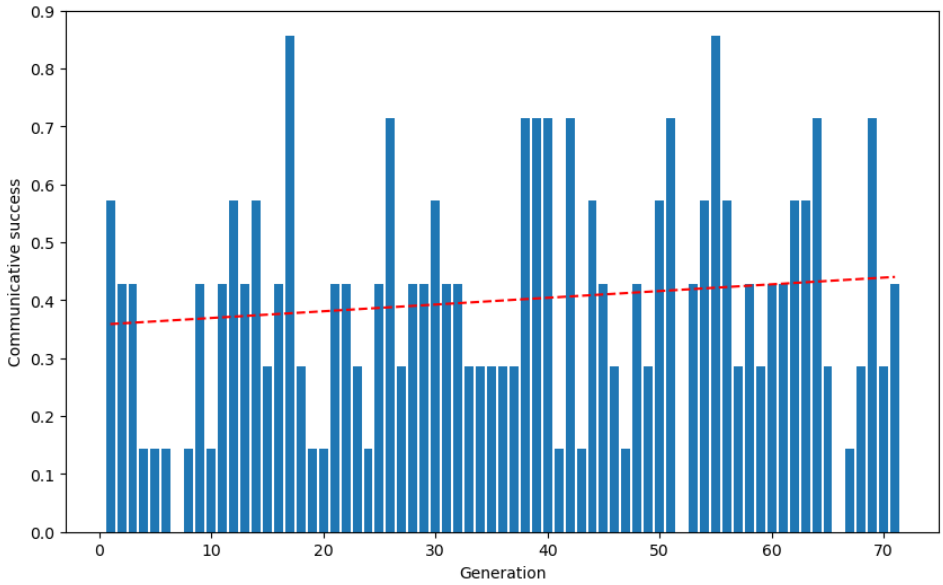



Figure 5: Plot of the first group CS; 29 participants, 16 y.o.

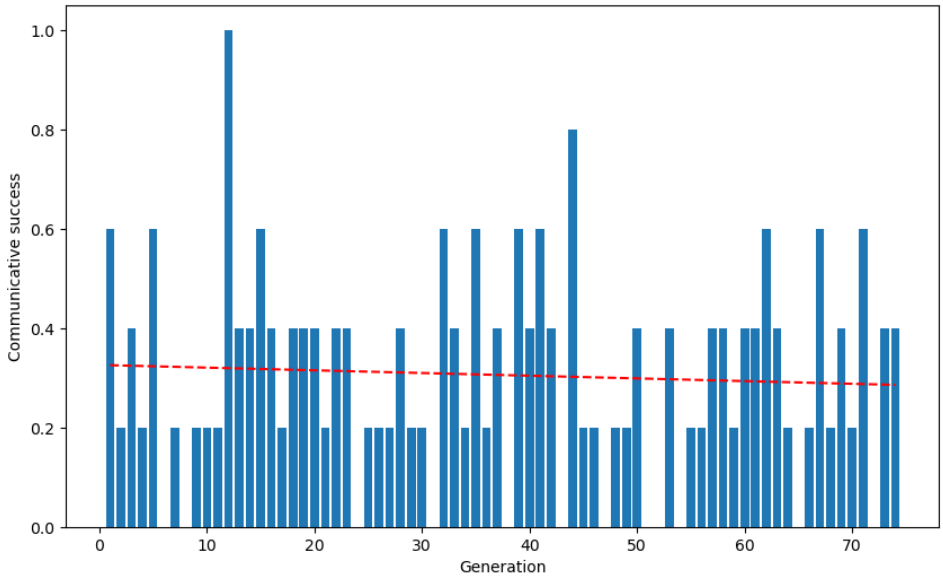



Figure 6: Plot of the second group CS; 20 participants, 12 y.o.

performance of the groups. Analysis of its causes is not in the scope of this paper. One could conclude, based on the presented Figures, that participants' age is a key factor shaping communicative success. The full data we have collected does not support this hypothesis. In our opinion, they are inconclusive at the time.

# 6 Summary

In the paper, we have presented a system that allows us to observe a process of artificial language emergence. Our implementation is customizable and suitable for children, requiring very little training. It also aims to provide minimal waiting time and detailed game course recording. In this section, we present plans for further development of the software and finish with some concluding remarks.

The most crucial improvement would be to substitute Server-Sent Events for regular WebSockets and shift away from the REST API architecture. SSE was extremely challenging to utilize, due to its lack of support and maintainability. In one instance, the Firefox browser classified our SSE emitter as a tracking script and completely blocked it, rendering the game inoperable. Such a scenario is intolerable and must be resolved as soon as possible.

During our tests, we encountered multiple instances of the front end displaying either no symbols or all symbols overlaid on top of one another, which is an undesirable behavior. To prevent this in the future, we believe that in addition to improving the frontend's handling of the data provided, or in this case, when no data is provided because the backend has not yet returned the required data, error handling and validation needs to be improved.

The color palette was not optimally chosen, and due to limited time, no AB testing was conducted for the interface. This resulted in diminished visual hierarchy and contrast, potentially leading to difficulties in certain scenarios. To address this issue, it is recommended to seek guidance from a UI/UX expert before the next system iteration to address this type of inconvenience and make the system more accessible and visually well-designed.

In the event of a sudden disconnection from the game, such as a browser or PC crash, the user must be manually restored. To do so, knowledge of the user's ID is required, and the process can be time-consuming, causing the game to nearly come to a complete stop. To address this issue, it would be advantageous to implement a system that links the user with their assigned cookies upon rejoining the game, thereby facilitating an automatic and seamless process.

After two game sessions, we presented the groups' results to the participants. To do so, we have created a plot, displaying correct and incorrect answers, generated from data queried via the `/game/{gameId}/admin/summary` endpoint. Although functional, manually utilizing a spreadsheet for this purpose is suboptimal. Therefore, it would be beneficial to automatically generate a graph for the specified games with a summary view.

Picto has proven useful in children's trials. Despite inconveniences caused by hardware malfunctions, we have managed to gather reasonable amounts of data. In our current work, we underestimated the influence of the system's resilience, and this is the main area for improvement. Future adjustments are intended to make

the software more user-friendly for researchers, but the core objectives are already fulfilled.

## References

[1] Nicolas Fay, Bradley Walker, Nik Swoboda, and Simon Garrod. How to Create Shared Symbols. *Cognitive Science*, 42:241–269, May 2018.

[2] Alicia Forsberg, Dominic Guitard, Eryn J. Adams, Duangporn Pattanakul, and Nelson Cowan. Working Memory Constrains Long-Term Memory in Children and Adults: Memory of Objects and Bindings. *Journal of Intelligence*, 11(5):94, May 2023.

[3] Bruno Galantucci. An Experimental Study of the Emergence of Human Communication Systems. *Cognitive Science*, 29(5):737–767, September 2005.

[4] Tao Gong, Andrea Baronchelli, Andrea Puglisi, and Vittorio Loreto. Exploring the Roles of Complex Networks in Linguistic Categorization. *Artificial Life*, 18(1):107–121, December 2011.

[5] Kenji Hakuta, Ellen Bialystok, and Edward Wiley. Critical Evidence: A Test of the Critical-Period Hypothesis for Second-Language Acquisition. *Psychological Science*, 14(1):31–38, January 2003.

[6] Vera Kempe, Nicolas Gauvrit, Alison Gibson, and Margaret Jamieson. Adults are more efficient in creating and transmitting novel signalling systems than children. *Journal of Language Evolution*, 4(1):44–70, January 2019.

[7] Wisal Khan, Teerath Kumar, Cheng Zhang, Kislay Raj, Arunabha M. Roy, and Bin Luo. SQL and NoSQL Database Software Architecture Performance Analysis and Assessments—A Systematic Literature Review. *Big Data and Cognitive Computing*, 7(2):97, May 2023.

[8] William Labov. *Language in the inner city: Studies in the Black English vernacular*. University of Pennsylvania Press, 1972. Issue: 3.

[9] Thomas F. Müller, James Winters, and Olivier Morin. The Influence of Shared Visual Context on the Successful Emergence of Conventions in a Referential Communication Task. *Cognitive Science*, 43(9):e12783, September 2019.

[10] Thomas F. Müller, James Winters, Tiffany Morisseau, Ira Noveck, and Olivier Morin. Colour terms: native language semantic structure and artificial language structure formation in a large-scale online smartphone application. *Journal of Cognitive Psychology*, 33(4):357–378, May 2021.

[11] Jonas Nölle and Bruno Galantucci. Experimental Semiotics: past, present, and future. preprint, PsyArXiv, September 2021.

[12] Limor Raviv, Antje Meyer, and Shiri Lev-Ari. The Role of Social Network Structure in the Emergence of Linguistic Structure. *Cognitive Science*, 44(8), August 2020.

[13] Catherine E Snow and Marian Hoefnagel-Höhle. The critical period for language acquisition: Evidence from second language learning. *Child development*, pages 1114–1128, 1978.

[14] Duncan J. Watts and Steven H. Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393(6684):440–442, June 1998.

[15] Julian Zubek, Michał Denkiewicz, Juliusz Barański, Przemysław Wróblewski, Joanna Rączaszek-Leonardi, and Dariusz Plewczynski. Social adaptation in multi-agent model of linguistic categorization is affected by network information flow. *PLOS ONE*, 12(8):e0182490, August 2017.

# The Autonomy-Driven Approach in Software Localization Teaching

Hubert Kowalewski*

## 1 Introduction

Despite the fact that software has been translated and adapted to various socio-cultural contexts for several decades,[1] translation studies were slow to recognize software localization as a legitimate topic of scholars' attention. This situation is gradually changing. Recent years have witnessed a number of publications on video games translation e.g. [2, 3, 4], as well as on more general issues relevant to software by and large e.g. [5, 6, 7, 8]. Given the fact that localization is a rather new topic in the scholarly literature, it is perhaps not surprising that the literature on teaching software localization is more scarce.

This is unfortunate, since software localization is in many important ways different from the "conventional" translation of artistic and everyday texts. This, in turn, means that the skills acquired by students during translation courses may not suffice when the prospective translators are faced with challenges specific to localization. Most of the challenges stem from two properties translatable phrases in software: unpredictability and decontextualization. This chapter outlines a general approach to teaching localization with special emphasis on the problems resulting from these two properties. Throughout this chapter I am going outline what I term the autonomy-driven approach (ADA) to teaching localization. As the name suggests, the main goal of the approach is to help students to develop professional autonomy, i.e. a set of skills necessary for the successful handling of a wide variety of potential problems with minimal guidance from other people involved in the process of software development. The autonomy-driven approach does not imply that localizers will be able to solve all potential problems on their own. On the contrary, in software localization close cooperation with other team members (programmers, graphic designers, etc.) is even more important than in conventional translation. Rather, the autonomy should be understood as an ability to make independent informed decisions in the face of a wide variety of unpredictable problems that may arise during localization.

In many respects the approach proposed in this chapter is consonant with

*Corresponding author — hubert.kowalewski@mail.umcs.pl

[1] For instance, one of the first comprehensive guides to localization, now largely outdated Bert Esselink's *A Practical Guide to Localization* [1], has been around for over two decades.

Dingfelder Stone's emphasis on importance of translator's autonomy [9] and Kiraly's view on translation competence as a multi-faceted skill emerging from complex interactions between the translator, the text, and various elements of broadly understood context [10, 11]. While Dingfelder Stone stresses the importance of authenticity in translation training, in Section 3.2, I will outline an approach slightly different than the author's, as I will allow that authentic material for localization is software specially adapted to classroom environment, even if it is not a fully-fledged market-ready product. I will only require that the material imitates the features of the market-ready well enough to reproduce the challenges that students will encounter during their future professional career. My understanding of the term *autonomy* is close to Kiraly's *empowerment* in that both involve equipping trainees with, as Williamson aptly puts it, "the agency and confidence to be fully-fledged translators" [see [12], p. 21]. The main difference between the two is that autonomy puts greater emphasis on the need to achieve an optimal equilibrium between localizer's self-reliance when faced with translation challenges and the awareness of the need to cooperate with other members of the development team. In other words, while students should definitely feel empowered in their prospective professional careers, they should also understand the technological and organizational constraints under which they operate.

In the following sections I will discuss some possible difficulties encountered by localizers during their work. I will focus primarily on the challenges arising specifically from the way software is localized rather than purely "linguistic" problems, i.e. the difficulties resulting from the semantics and grammar alone. I will also resist the temptation to explicate the difference between the terms *translation* and *localization*.[2] Instead, I will follow the well established convention of reserving the latter term for the process of adapting software to a specific linguistic, social, and cultural context (which involves, but is not limited to, translating texts displayed to users) and the former term for the process of rendering texts written in a source language into a different target language [2, 4].

## 2 The autonomy-driven approach — basic assumptions

Both conventional translation and localization require taking decisions influenced by many factors. Oftentimes, the decisions are hard, because the relevant factors "pull in different directions". Take, for example, a question of whether the word *cancel* displayed on a clickable button inside a graphic user interface should be translated into a target language as an imperative or an infinitive. Arguably, this seemingly simple decision may be affected by a number of pragmatic, stylistic, and even cultural factors in the target language. While the localizer may consult the software developers about the preferred form, the developers may not be aware of all the linguistic nuances, especially if they are not familiar with the target language. More often then not, the localizer is the only person in the team competent enough to make this decision and needs to make it without much additional help and guidance. What is worse, it is usually impossible to predict in advance what

[2] Readers interested in this subject matter will find competent discussions in Dunne [5].

kind of software the localizer will work with and, consequently, what kind of problems they will face. Translatable phrases for a text user interface (TUI) are subject to different constraints than phrases for a graphic user interface (GUI). Graphic interfaces designed for desktop computers are different from interfaces designed for smartphones, virtual reality goggles, augmented reality devices, etc. Thus, the skills acquired by localizers during a course focused primarily on the localization of GUIs for smartphones may be largely irrelevant when it comes to localizing software with TUIs. For this reason, the autonomy driven approach emphasizes the need for creative and flexible application of the tools at localizer's disposal rather than mere efficiency of localizing software of certain pre-defined format.

For this reason, it is crucial to familiarize students with tools implementing the principle of separation of translatables from the source code, like the localization system GNU discussed in the sections to follow, rather than, for example, tools allowing for localization directly in software's GUI. Even though Leiva and Alabau [13] observe that localizers tend to perform better when translating directly in the GUI rather than in plain gettext, this conclusion is more relevant as a recommendation for IT companies who need to optimize the workflow within the development team rather than for teachers designing courses on localization. Arguably, the reason why localizers perform better when working directly in the GUI is that it is easier to render translatables in the target language when the relevant visual context is in place. Undoubtedly, the presence of relevant context facilitates localization, but it does not necessarily promote the creativity, imagination, and flexibility required during localization of "contextless" strings extracted from software's source code are gathered in a separate directory. On the other hand, a student proficient at working with "contextless" strings should find it relatively easy to localize with the GUI in place. In other words, while industry executives may want to remove obstacles from the process of localization in order to promote localizers' performance, teachers may want to leave the obstacles in place in order to show how the obstacles can be handled. As a result, the prospective localizers may be better prepared for successfully dealing with potential problems in their future professional career.

The autonomy-driven approach is heavily influenced by the so-called communicative language teaching (CLT) — and methodological approach in foreign language teaching [14]. Even though CLT disfavors translation (let alone software localization) as an effective way of mastering a foreign language,[3] CLT and ADA share the general philosophy of teaching. Both of them aim at developing a set of practical skills that help students to perform successfully in various unpredictable real-life situations. This kind of competence is viewed as a practical skill rather than a unreflective application of formal rules or abstract guidelines. More specifically, in foreign language teaching, linguistic competence is viewed as a tacit skill of producing and understanding utterances in a given language rather than an ability to apply explicit rules of grammar. In ADA, localizer's competence is viewed as a practical ability to adapt translatables to a foreign linguistic, cultural, and social environment rather than a declarative knowledge of various rules and guidelines governing the way in which translatables are to be rendered in the target language.

[3] See, however, Kocbek [15] for a reappraisal of translation within a communicative method curriculum.

# 3 The main principles of the autonomy-driven approach

## 3.1 Focus on decontextualization and unpredictability

One difference between conventional translation and localization is the scope of the material falling within the purview of the process. As many authors rightly note e.g. [16], localization involves adapting many elements of software that are not, strictly speaking, "translated," at least not in a typical sense of this term. This includes, but is not limited to, the adjustment of calendar dates, currency formats, and conventions of mathematical notation (e.g. the character used as the decimal separator). Yet from the point of view of a translator and a localizer, another difference is perhaps more important and troublesome. In conventional translation, artistic and everyday texts are stable and linear entities, while translatables in a directory do not have a fixed default order. The fixed arrangement of sentences in a conventional text constitutes a rich linguistic context for the interpretation of each sentence. At any point in the text the translators knows what information has already been provided to the reader and what information is still to be delivered. This is clearly not the case for translatable strings extracted from software's source code. It may be the case that the user will see certain strings first (e.g. the name of the program on a splashscreen while the program is booting) and certain string last (e.g. *Game over* once the game is finished), but the order of an overwhelming majority of translatables is highly unpredictable and depends crucially on the actions of the user. From a localizer's point of view, there is typically no stable context, no pool of information that the user can be safely assumed to possess, and no predefined sequence of actions leading from one translatable string to another. This makes the process of localization radically different from the process of regular translation and requires dramatic adjustment of techniques and practices.

The separation of translatables from the rest of the codebase is presently a standard practice, although the formats in which the translatables are delivered to localizers vary. Typically, the localizer works with a structured directory of phrases and does not see where and how the string under consideration will appear until the program is actually run. This results in the decontextualization of translatables. Decontextualization is particularly problematic for software with GUIs, since the complex interdependence of textual and non-textual components of the interface, the distribution of translatables on the screen, limitations of space, etc. cannot be always inferred directly from the directory of translatables. Nonetheless, the challenge is also present when text user interface software is localized, although the problems may be less severe. Unpredictability, in turn, pertains to the degree to which the eventual shape of the string presented to the user cannot inferred from its shape in the directory. This is typically the case with strings containing placeholders for variables, especially when the values of the variables affect the grammatical form of translatables. Since decontextualization and unpredicability tend to be the most challenging for students, they should receive relatively much attention, they should be covered early on during the course, and teaching materials should expose students to a number of problems resulting from these properties.

Common examples of unpredictable translatables are phrases with placehold-

ers for variables, phrases with singular and plural variants (both of the types are discussed in more detail in Section 4.1), and phrases whose inflectional form depends on the factors that cannot be inferred from the directory of translatables. An example derived from the game *Pyrates* [17] is the message in (1)(a), displayed when the player finds a chest on a treasure island, and its possible translations into Polish (1)(b) and (1)(c).

(1) (a) *You **found** a sturdy yellow chest with a steel padlock.*
(b) ***Znalazłeś** mocną żółtą skrzynię ze stalową kłódką.*
(c) ***Znajdujesz** mocną żółtą skrzynię ze stalową kłódką.*

As far as linguistic meaning is concerned, (1)(b) is the most accurate rendering of (1)(a); it is virtually as word-by-word translation, as much as the grammatical differences of English and Polish allow. The problem is that while the English past tense forms do not mark the grammatical gender of the addressee, the Polish past tense forms do. Consequently, the masculine verb form *znalazłeś* '(you) found' in (1)(b) is grammatically correct when the addressee is a man, but it is incorrect when the addressee is a woman. In other words, the proper form of the Polish equivalent of (1)(a) varies depending on the gender of the user, which, of course, cannot be inferred in any way from the directory of translatables. One could argue that referring to unspecified or generic addressees by means of masculine forms is a widespread phenomenon in Polish and adult female speakers do interpret such masculine forms as referring to them [18]. Nonetheless, the gender-biased translation in (1)(b) hardly contributes to optimal user experience when the user is a woman and the string calls for a different approach. One way to avoid the bias is to rephrase (1)(b) into the present tense, as in (1)(c). Since Polish present tense does not mark the gender of the addressee, the problem of the user's unknown gender disappears.

The rendering of a source text into a target language performed during localization is subject to different constraints than the rendering of the text in the process of translation. In fact, it is debatable whether the localization of highly decontextualized and highly unpredictable texts should be considered as a type of translation at all. Perhaps localization is more adequately characterized as an aspect of designing user experience for a foreign linguistic, cultural, and social context. Notice that in principle, the gender-biased (1)(b) is not a "bad" translation of (1)(a) in any obvious sense; in fact, it may be a perfectly adequate in a literary text where the addressee is a man. Yet (1)(b) is clearly a poor instance localization, since, in principle, the gender of the user cannot be determined in advance. In other words, what makes (1)(b) a potentially adequate translation and an unavoidably inadequate localization is that during translation the translator may sometimes know the gender of the addressee and use the correct verb form; yet during localization the localizer cannot know in advance the gender of the user. Thus, while a competent translator may simply demand more context to adjust the past tense form, a competent localizer realizes that the relevant context simply cannot be determined.

This example demonstrates that competent localization is a creative process and required much flexibility. Even if one formulated a provisional rule like "When localizing into Polish, change past tense forms into present tense forms to avoid gen-

der bias," it is not a kind of rule that can be taught explicitly during general course on localization, not even in Poland. The most important reason for that is that if the rule were formulated as a general prescription, it would fail to apply across the board. In some cases preserving the past tense is be a better solution. Consider, for example, the message like *5 files were deleted*. Here, the crucial information is that the process of deleting the file has been completed and rendering the message in the present tense would falsely suggest that the process is currently in progress. So when exactly should English translatables in the past tense be rendered as Polish phrases in the present tense? Unfortunately, the scope of application of this putative "rule" cannot be determined in advance. The only useful answer would be "When the need for preserving the original tense is overridden by other important factors, like the desire to avoid biased language." However, recognizing the situations where the rule of this sort should be followed is not a matter a following yet another rule. It requires cultural and social competence to realize what kind of problems a particular phrase may cause in a particular socio-cultural situation, as well as empathy and imagination for predicting and preventing potential problems with comprehension on the part of the players. The teacher should help students to develop this kind of multi-faceted competence, but the competence cannot be codified in strict explicit rules.

## 3.2 Authenticity

The need for developing this socio-cultural competence is one of the reasons motivating the use of authentic material for classroom activities. What counts as authentic in translation/localization teaching is somewhat debatable (see Pacheco Aguilar [19] for a brief overview of various possible senses of this term). My notion of authenticity is similar to the one adopted in the communicative language teaching, where students practice the use of language in *simulated real-life* situations. They may, for instance, role-play ordering a dinner in a restaurant or booking a hotel room. While the classroom activities are artificial in the sense that they are designed specifically for developing a particular element of linguistic competence, the activities should resemble real-life situations as closely as possible. In the context of localization, ADA advocates localizing software usable for real-life purposes. This does not necessarily mean that the only admissible software for in-class localization is the software actually used outside the classroom. It only means that the software used for localization should have a potential function relevant for a real-life users in real-life situations. For example, if the software to be localized is a text editor, the text editor does not have to be actually used outside the classroom, but it should be possible to use the program to edit a text.

This guideline discourages the use of what could be termed the "hello world" software, i.e. programs created specifically to demonstrate some technical aspect of software design, but with no interesting function for end users. Even though "hello world" programs may be useful tools for teaching programming, they may be too contrived to demonstrate the challenges that prospective localizers are likely to face in their professional careers. An analogy to foreign language teaching may help to clarify this point. Consider a mock-up conversation between foreign language students in which one plays the role of a waiter and the other a restaurant customer.

The aim of this activity is to practice the production of linguistic expressions used during an actual visit to a restaurant. This kind of practice may be effective despite the fact that the classroom is not an actual restaurant and that the students are not actual waiters and customers. The classroom arrangement is make-believe, but it approximates an authentic situation in many relevant aspects, so that and the skills exercised during the activity can be used in real-life circumstances. This stands in sharp contrast to, for example, the grammar-translation method consisting largely of studying explicit grammatical rules and using them to translate sentences. This kind of classroom arrangement is equally artificial, but does not attempt to approximate real-life acts of communication in a foreign language. By the same token, in ADA it is not necessary to produce authentic, market-ready localization of authentic market-ready software, as long as the programs to be used for localization successfully imitate important aspects of authentic programs.

## 3.3 Test early, test often

The autonomy-driven approach stresses the need for frequent testing of localized phrases. Apart from the obvious need for checking whether certain solutions work in a running program, frequent testing helps students to overcome difficulties resulting from decontextualization and unpredictability of translatables. In the case of phrases with placeholders for variables, it is during testing that students have a chance to see the final form of a string will have once the variables are in place. It also allows students to see the texts in the proper context, i.e. inside a running program. I will return to the discussion on the need for frequent testing in Section 4. At this point, I will limit myself to a cursory remark on aspects of classroom setup facilitating early and frequent testing.

The need for frequent testing favors software written in interpreted languages. From students' and teachers' points of view, the optimal arrangement is when the effect of localization can be seen in the functioning program almost immediately after the phrase is rendered in the target language. The time necessary for compiling the entire source code before the program can be executed slows down the testing process and introduces an extra step (the compilation), which adds unnecessary complexity to the task. Even though in gettext, the translation system discussed in more detail in the following section, the files with localized strings do require compilation, the compilation can be performed swiftly. For instance, Poedit (the GUI editor of translatable phrases prepared with gettext) has a functionality of compiling the directory of translatables in the background while saving the modified file. Along with other utilities (e.g. a Windows point-and-click Python launcher for software written in Python), it can offer student a convenient point-and-click experience which entirely eliminates the need for any command-line tools. This is particularly beneficial for students who have little or no experience with command-line programs and software development in general.

## 3.4 Engagement

As already mentioned, ADA advocates prioritizing the challenges caused by unpredictability and decontextualization during the teaching process. Localizing

decontextualized and unpredictable translatables may pose serious challenges to students with no prior experience in software development and therefore the proper in-class setup should pose as few additional difficulties as possible. A lot depends on the prior knowledge of students about various types of software, but as a general rule, types of software that may be unfamiliar to students should be avoided. While transcompilers may require localization, they are a poor choice for students who do not know what compilation (let alone transcompilation) is. Web browsers and text editors are better in this respect, since students can be reasonably expected to have rich experience with this types of programs, but these programs also run the risk of featuring specialized or technical vocabulary. The need for finding the equivalent terms in the target language for terms like *widows* and *orphans* as used in a text editor may require students to learn about the intricacies of typesetting, which is likely to needlessly distract them from the main topic of the course. Simple computer games, especially ones devoid of complex terminology, are a safer choice for most purposes.

Computer games are a relatively safe type of software, as students can be reasonably expected to be familiar with this kind of programs, even if they are not avid players. Games have another advantage over software with more "practical" applications like transcompilers, web browsers, and text editors: they are more engaging and more attractive to interact with. The potential of engagement is a useful property of the material for in-class localization not only because it makes classroom activities more fun, but also because it encourages active exploration into the program before localization proper even begins. Having prior experience with software greatly facilitates the localization of decontextualized and unpredictable translatable. The localizer may simply recognize the phrase in the "contextless" directory as something appearing in a particular part of the program's interface and may therefore have a decent idea about how the phrase works within the program. Yet even when the localizer does not immediately recognize a translatable, having a general idea about the layout of the interface and the way translatables are presented to users may help to find an adequate equivalent in the target language. For this reason, it is highly advisable to ask students to run and use the program localized during in-class activities before localization proper begins. Arguably, the teacher may also ask students to browse through the interface of a transcompiler, a web browser, or a text editor in order to familiarize themselves with the program's interface, but this activity would be somewhat forced, unnatural, and tedious. It would promote carelessness and boredom rather than spontaneous engagement and, as a result, it would sabotage the main goal of the preliminary stage in which localizers are expected to "get to know" the program. On the other hand, using the programs purposefully, i.e. for transcompiling source code, browsing networked resources, or editing texts, forces students to focus on the actual function of the program and distracts them from the translatables. In a properly designed TUI computer game, the world in the gameplay may be explored chiefly through the texts appearing in the interface.[4] Therefore, purposeful engaged interaction with the software coincides with familiarizing oneself with texts that will eventually wind up in the directory of translatables.

[4]This, of course, means that a properly designed game needs to have rich textual content and cannot be played effectively solely by interacting with non-linguistic graphic elements.

# 4 Teaching localization — practical issues

This section focuses on the specific challenges stemming from the unpredictable and decontextualized character of translatables. I will illustrate these challenges with phrases derived from the program *Pyrates* [17]. *Pyrates* is a suite of four TUI games. The games are written in Python. Since Python is an interpreted language, it is possible to test the localized strings quickly and easily at any stage of classroom activities, because the games do not need to be pre-compiled before launching (see Section 3.3). While TUI programs tend to be somewhat less attractive for users (which may negatively impact the potential for engagement discussed in the previous section), they have a number of advantages which make them useful teaching tools in some important respects. For our purposes, two advantages are particularly important. Firstly, TUI programs do not distract students with overly eye-catching graphics, which helps students to focus on the localized texts. Secondly, TUI is more "unforgiving" as far as various types of imperfections in the translatables are concerned. While in graphic user interface double space, missing line breaks, or excessively long lines of text may not stand out from the mass of visual elements, in text use interface they are typically very noticeable. For example even a single superfluous space can cause very conspicuous misalignment of texts. This "unforgiving" character of TUI promotes attention to detail and helps to make students aware that (contrary to the frequent misconception held by learners) localization is not always a matter of rendering linguistic content alone; it is equally important to pay attention to how the phrases behave in the interface. This, in turn, demonstrates the importance of testing. A phrase which may "look good" in the directory of translatables does not always "work well" when the program is run and the exact nature of the problem becomes apparent only when the program is executed.

*Pyrates* are localized by means of the software localization system GNU gettext. In gettext, translatables are marked in the source code of the program (the stage is commonly referred to as *internationalization*) and then extracted into a file called a portable object template (POT). The POT is then processed into a portable object file (PO), which contains a list of human-readable translatable phrases. A sample entry from the Polish PO file of *Pyrates* is presented in [2]. The text immediately following the "`msgid`" tag is the original phrase derived from the source code and the quotation marks immediately following the "`msgstr`" tag is where the localized phrase is placed. The line starting with # is a comment reference to the location of the translatable phrase within the source code and (optionally) additional information.

(2)
```
#: modules/isles.py:151
msgid ""
"You find Cofresí's message in a bottle; it contains misleading information: \n"
msgstr ""
```

Localizers work mostly with PO files, but the exact scope of their responsibilities depends on the organization of workflow within the development team. After completing the localization in a PO file, the file is compiled into a machine object file (MO) usable by the program. After the MO file is placed in the appropriate folder within the project and the program is configured to use the translatables

```
********************************** EN ROUTE ***********************************
Ch: 0/5                                                  F: 10, G: 0, R: 5, K: 5

You find Cofresí's message in a bottle with encrypted parts of island names:

Log book entry November 05, 1822 (Tuesday) 18:59:28
 - *202…   TREASURE ISLAND
 - }2|$…   PIRATES' ISLAND
 - ?37$…   TREASURE ISLAND
 - #|$9…   DEFUNCT DISTILLERY

  [1] Buhejeri    (1 day away)
  [2] Koce        (3 days away)
  [3] Sheje       (4 days away)
  [4] Yulupu      (2 days away)
                                                          [Z] exit game
Where to now?
```

Figure 1: Phrases with plural forms

from a particular MO file, the executed program uses the localized phrases instead of the phrases hardcoded in the source code.

## 4.1 Unpredictability

Gettext includes several built-in functionalities that help to effectively deal with the inherent unpredictability of certain types of translatables. Thus, gettext usually allows for building grammatically and stylistically correct phrases even when the phrase is put together from several unpredictable building blocks. This, of course, is very welcome from user's point of view, but it requires competence and good intuition on the part of the localizer. *Pyrates* are designed specifically to expose the localizers to a wide variety of unpredictable phrases. For example, the games in the suite feature as much as four different types of placeholders for variables: positional variables with curly brackets ({}), keyword variables with curly brackets ({`variable`}), keyword variables with $ prefix ($`variable`), and keyword variables with % prefix (%`variable`). This diversity of placeholder types is not justified by the complexity of the program; its sole role is to teach students how to work with variables delivered in different formats.

One type of unpredictability results from the fact that some translatables include numeric variables and the grammatical form of other elements in the translatable depend on the value of these variables. In many languages, this is simply a matter of selecting the singular or plural form of a word depending on the values of the variables appearing in the phrases. In *Pyrates*, one example is the phrase showing the distance from the current location of the player to an island (Figure 1).

In gettext, the phrases of this sort can be generated by specifying two versions of the original phrase, one with the singular ("`msgid`") and the other with the plural ("`msgid_plural`") form depending on the value of the variable in the placeholder {}. Localizers are offered several localizers `msgstr` tags ("`msgstr[0]`", "`msgstr[1]`", etc.) for the plural form phrases in the target language (see [3]).

(3)
```
#: pyrates.py:187
msgid "({} day away)"
msgid_plural "({} days away)"
msgstr[0] ""
msgstr[1] ""
msgstr[2] ""
```

Since the variable placed in {} has a specific value once the program is executed, it is possible to select the correct grammatical form of the word *day* for the entire phrase. This means that the awkward phrasing like "({} day(s) away)" can be avoided. This solution offers a optimal user experience for the player, but also requires the translator to figure out which variant of the localized phrase should be placed in each of the msgstr tags. While in many languages this is simply a matter of supplying the singular form in `msgstr[0]` and the plural form in `msgstr[1]`, Polish has two variants of plural noun forms: the nominative case is used for quantities within the range 2-4 (e.g. *dwa* ***koty*** 'two cats') and the generative case for quantities within the quantities greater than 4 (e.g. *pięć* ***kotów*** 'five cats').[5] Depending on the peculiarities of the target language, this functionality of gettext may require more instruction from the teacher. Native speakers are obviously highly proficient in their mother tongues, but the proficiency is a type of tacit knowledge (in Polanyi's [20] sense) and students may not be fully aware of various grammatical nuances. For example, native speakers of Polish tend to be rather surprised to realize that Polish requires either the nominative, or the generative case in plural sentences (depending on the exact number), even though they use these forms correctly on everyday basis.

Another type of unpredictability problematic for localizers is a consequence of how gettext handles multiple occurrences of identical translatables throughout the source code. By default, such multiple occurrences are combined under one entry in the PO file. For example, in *Pyrates* the phrase "Huh...?" displayed when the player inserts an invalid input appears as many as 27 times. Yet in the PO file there is only one entry for the phrase (with 27 references to the source files in the comments), rather 27 separate entries for each occurrence of the phrase. In general, this is a desirable behavior, since it prevents the unnecessary bloating of the directory and saves localizers' time (they need to localize only one phrase instead of 27). However, the default behavior may be undesirable when identical phrases appear in different contexts. Consider, for example, the phrases displayed in *Pyrates* when the player scavenges on a treasure island (Figure 2). The message about a chest found (*You find a STURDY BROWN chest with a BRASS padlock* in Figure 2) is built from the translatable phrase in (4)(a):

(4) (a) "You find a {attribute} color chest with {padlock} padlock."
(b) "{attribute} {color}, {padlock} padlock: empty"

The elements inside the curly brackets are variable placeholders replaced by other translatable phrases from the PO file, i.e. a descriptive adjective about the

[5] The system of plural noun forms in Polish is more complex, but the details are irrelevant for this discussion.

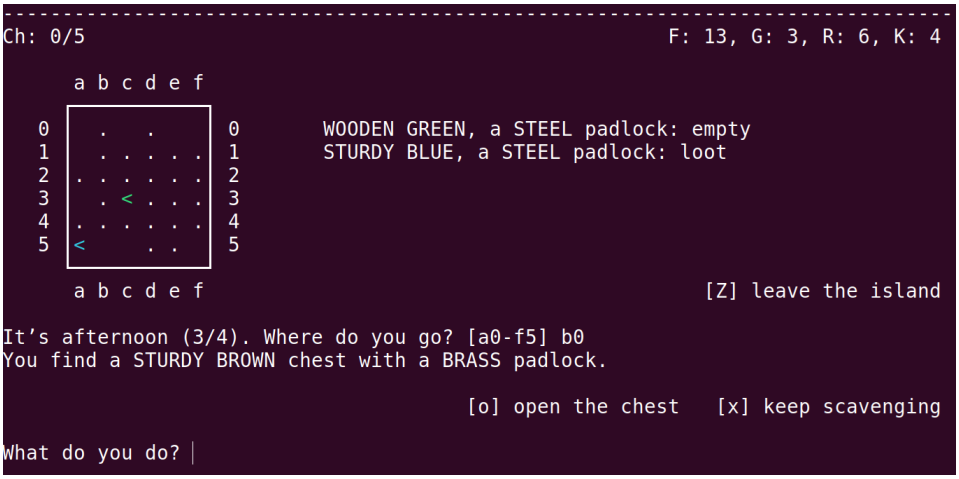



Figure 2: Phrases in different contexts

chest, a color adjective, and an adjective describing metal of material respectively. The adjectives are also used to build the entry in the chest list displayed next to the island map on the left. The translatable serving as the template for the list entry is shown in (4)(b).

While in English the adjectives like *sturdy*, *red*, and *iron* have the same grammatical form in both the sentence about the chest and the item in the chest list, other languages may require different grammatical forms in these two contexts. For example, in Polish adjectives are subject to declension and it may be more natural to use a nominative form in the chest list next to the island map and some other case in the message about a chest found.[6] In order to supply the two different grammatical forms of the adjective *brown*, the localizer would need to have two separate entries in the PO file: one for the *brown* in the sentence and the other for the brown in list item. However, gettext by default combines the two occurrences of the word *brown* into a single entry in the PO file.

This problem is solved during the internationalization stage by marking the separate occurrences of the word *brown* in the source code as occurrences in different contexts. After extraction into a PO file, the occurrences are presented as two separate entries with additional "`msgctx`" tags clarifying the context. This solution effectively prevents gettext from combining two identical phrases into a single entry and allows localizers to supply adequate grammatical forms for each context.

[6]The exact grammatical case depends on the exact form of the sentence, but a transitive sentence with the accusative is a likely candidate, e.g. Znajdujesz MOCNĄ BRĄZOWĄ skrzynię z MOSIĘŻNĄ kłódką.

(5)
```
msgctxt "chest list"
msgid "GREEN"
msgstr ""

msgctxt "chest found"
msgid "GREEN"
msgstr ""
```

When used skillfully, the functionality of separating contexts is a powerful tool for generating grammatically and stylistically correct phrases in the target language, a behavior much appreciated by end users. Yet the functionality also increases the complexity of localizer's work. With separate contexts, it is no longer enough to know the linguistic meaning of the source phrase in the target language. It is also necessary to know how the phrases from each context are displayed to users and how the final message is put together from other constituents. While in principle the way in which gettext puts phrases together is entirely predictable and deterministic, in practice it is usually difficult to guess the final shape of the phrase displayed during runtime on the basis of the list of translatables alone.

From the point of view of the teacher, this is a good opportunity to emphasize the importance of testing. It is worth explaining that testing is not merely a way of making sure that "everything is right" with the localization, but also that it is a way of getting a better understanding of how the program works. This is particularly important for localizers who do not actively contribute to the creation of the source code or who are not software developers at all. For such localizers, the internal mechanisms of the software under localization are typically rather mysterious, if not downright puzzling. Fortunately, with several iterations of the localize-test-correct cycle, even such localizers should be able to propose adequate phrases for all contexts.

## 4.2 Decontextualization

As already mentioned, testing is important, because an optimal way of localizing a phrase becomes apparent only when the phrase is seen in its natural context. As Bernal-Merino writes about the localization of computer games:

> The fact that translators work from spreadsheets, which is almost always the case, with no access to the actual graphics or to the moment in the gameplay where these graphic-embedded words can be seen, makes it more difficult to produce the ideal translation (...) Some constraints remain undetected until the localisation process is finalised and rushed for gold copy (the master copy) and international release. [[2], p. 131]

Even though it is somewhat debatable whether in 2015 it was really "almost always the case" that localizers had no access to graphics (what about open source games in which all graphics is publicly available?) and whether it is "almost always the case" that they worked from spreadsheets (what about games localized via gettext, XML, or JSON files?), the author correctly captures the problems re-

lated to the lack of adequate testing before the release.[7] This problem is sometimes exacerbated by erroneous assumptions about localization held by people with background in conventional translation. Undoubtedly, competent translators proofread their texts before sending them back to their clients. Yet many translators-turned-localizers believe that the equivalent of the proofreading stage in the process of localization is re-reading the phrases in the directory of translatables. This is clearly not the case. The equivalent of proofreading is running the program with the localized phrases. Unfortunately, it may also be difficult for teachers to weed out their students' habit of "proofreading by re-reading" rather than "proofreading by testing."

This perhaps should not come as a surprise, since the lack of adequate testing may also happen to seasoned and experienced localizers. Figure 3 juxtaposes screenshot from *Adrift*, a small survival game from the *Pyrates* suite. The upper half of the figure shows original English phrases and the lower half shows French localized phrases submitted by a user via Weblate.org.[8] As far as the linguistic correctness is concerned, the translatables have been localized properly, yet some "non-linguistic" problems are evident. The most apparent shortcoming is the misalignment of the the two-column legends on the right of the square ASCII-art diagram. In the original English version the legend and the list of keystrokes in the square brackets is organized into orderly left-aligned columns. The alignment results partly from the spaces in the translatable phrases; the corresponding snippet of the PO file is presented in (6).

(6)
```
#: adrift.py:265
msgid "{}  bottle       {}  shark"
msgstr "{}  bouteille       {}  requin"

#: adrift.py:267
msgid "{}  sailcloth    {}  driftwood"
msgstr "{}  toile à voile    {}  bois flotté"

#: adrift.py:269
msgid "{}  food         {}  chest"
msgstr "{}  nourriture         {}  coffre"

#: adrift.py:271
msgid "{}  the raft     {}  ocean"
msgstr "{}  le radeau     {}  océan"
```

Notice that when the left-hand words next to the curly-bracket placeholders are translated into a target language, the spaces between the words shift the right-hand

[7] Although the fact that in some projects "some constraints remain undetected until the localisation process is finalised" probably suggests poor organization of workflow within these projects rather than a difficulty inherent in the very process of localization.

[8] Weblate.org is an online translation management system (TMS) for continuous localization, which allows registered users to submit translations to projects of their choice. In continuous localization, files with translatable phrases are hosted separately in a TMS and linked to the main repository of source code through a version control system cf. [21].

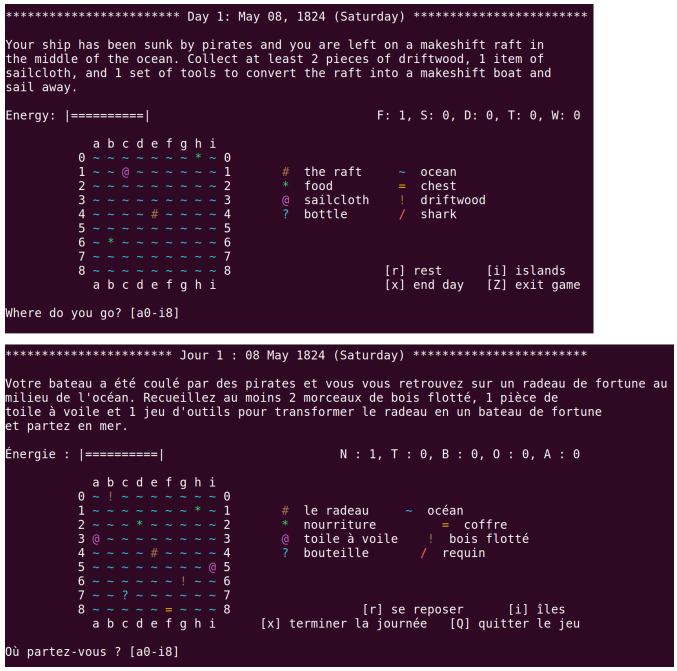



Figure 3: Screenshots from *Adrift* with English and French phrases

words relative to the words in the lines above and below. This happens when the number of characters in the translated words differs from the number of characters in the original word. In order to preserve the orderly alignment of the columns, the number of spaces between the words needs to be modified. In principle, it is possible to calculate the number of spaces required for each translatable on the basis of the number of characters in the original English phrases and the number of characters in the localized phrases, but the work is quite tortuous, abstract and complex. A more intuitive way is to test a work-in-progress version of the localization to see how the menu is aligned during runtime. The way in which the legend is misaligned in Figure 3 gives the localizer a good idea of how many spaces should be inserted into the translatable and even if the initial estimate is incorrect, further rounds of testing should make it possible to arrive at a satisfying solution. If so, why are the legends in the French version misaligned?

We may only speculate about the actual reasons, but it seems likely that the localizer did not test the French phrases. This is perhaps due to peculiarities of working in online translation management systems. An online TMS is a very convenient tool for managing workflow in a development team. It typically does not require the localizers to work with PO files directly, since translatables are accessed via a specialized frontend in a web browser. This also means that localizers do not need to have access to the entire source code necessary to execute the program. In fact, in many projects under propriety licenses the access to the entire codebase is deliberately restricted. This means that localizers may not have the chance to test the localization even when they are willing and competent enough to do

this. Note, however, that in the case of *Adrift* the source code is publicly available, so the localization could be tested if the localizer wished to do so, although that would require some extra work. Notice also that the numbers of spaces between the words in the French versions of the phrase are the same as the numbers of spaces in the original English phrase. This suggests that the localizer simply decided to preserve the original spacing in the localized phrase, perhaps without being aware that this solution is going to result in misalignment. In many cases, preserving the original spacing patterns of the source phrases (especially easy to overlook phrase initial and phrase final spaces) is a sensible decision. Here, however, the result is far from satisfying, but the exact nature of the problem this is evident only when the phrases are displayed during the gameplay and much less evident when the phrases are investigated in a directory of translatables.

A less apparent problem in the French version of *Adrift* is the date strings at the top of the screenshots: the English version of the phrase is "Day 1: May 08, 1824 (Saturday)" rendered as "Jour 1 : 08 May 08 1824 (Saturday)." The French localization mixes the French word *jour* with English words *May* and *Saturday*. To see how this problem arises, let us look at the relevant snippet from the French PO file in [7].

(7)
```
#: adrift.py:240
msgid " Day {}: {:%B %d, %Y (%A)} "
msgstr " Jour {} : {:%d %B %Y (%A)} "
```

From the technical point of view, the date string is generated on the basis of a Python datetime object and a translatable phrase with % prefixed variables determining what information from the datetime object is going to be displayed in the interface. In this case, the fault lies with the variables %B and %A displaying the English names of the month and weekday respectively. In *Adrift* and other games from the suite, localizers cannot "force" the program to display non-English names, but the variables %B and %A can be safely removed from the localized phrase without triggering any runtime error.[9] Unfortunately, it is quite difficult for a non-programmer to predict how the cryptic variables are going to be rendered in the game on the basis of the PO file alone. Once again, the best way to find the optimal localized version is to see how the string is displayed during the gameplay and which variables are responsible for the English names of months and weekdays. To be fair, it should be noted that the French localizer did modify the order to the variables and removed the comma, so that the date string conforms to the French conventions more closely than the original English version, but the localizer was probably unwilling to remove any variables from the translatable. In general, this is a sensible strategy, since in most cases removing variable placeholder frequently leads to incomplete phrases (in the best case scenarios) or to fatal runtime errors (in the worst case scenarios). Here, however, localizers enjoy greater flexibility and since some variables are inevitably filled with English words, the best solution is to remove the variables altogether.

[9] It is also worth adding that this information is included in the documentation of *Pyrates* in the section about localizing dates.

# 5 Conclusion

Classes on software localization are an important addition to courses on practical translation, because many challenges during localization are quite different from the challenges during conventional translation of texts. In order to prepare prospective localizers for these challenges, problems specific to software localization should be addressed explicitly. In particular, the skills required for conventional translation are typically not enough to successfully handle unpredictable and decontextualized phrases. Also, the adequacy of localization cannot be evaluated on the basis of "proofreading" the directory of translatables, even though proofreading a translated text is usually enough to evaluate the adequacy in conventional translation. The equivalent of proofreading is testing localization within a running program, since only them the localized phrases appear in the context in which they will be presented to users.

The teacher cannot predict in advance what kind of software the students will localize during their professional career. Perhaps the students will they work mostly with messages printed in a text user interface or elements of a graphic user interface, or texts overlaid onto physical objects in an augmented reality display. Since the constraints under which prospective localizers will have to work are largely unpredictable, it is impracticable to teach localization for fixed and highly specific types of user interfaces. A localizer highly proficient at localizing graphic user interfaces for smartphone applications may be almost entirely helpless when it comes to localizing a texts for text user interfaces or augmented reality displays. Thus, it is more effective to seek to develop students' autonomy, i.e. the ability to use available tools creatively and flexibly depending on specific problems arising in specific projects.

# References

[1] B. Esselink, *A practical guide to localization.* in The language international world directory, no. 4. Amsterdam: John Benjamins, 2000.

[2] M. Á. Bernal-Merino, *Translation and localisation in video games: making entertainment software global.* in Routledge advances in translation studies. New York: Routledge, 2015.

[3] C. Mangiron, "Reception studies in game localisation: Taking stock," in *Benjamins Translation Library*, vol. 141, E. Di Giovanni and Y. Gambier, Eds., Amsterdam: John Benjamins Publishing Company, 2018, pp. 277–296. doi: doi:10.1075/btl.141.14man.

[4] M. Deckert and K. Hejduk, *On-Screen Language in Video Games: A Translation Perspective*, 1st ed. Cambridge University Press, 2022. doi: doi:10.1017/9781009042321.

[5] K. J. Dunne, Ed., *Perspectives on Localization.* Amsterdam-Philadelphia: John Benjamins Publishing, 2006.

[6] K.-D. Schmitz, "Indeterminacy of terms and icons in software localization," in *Terminology and Lexicography Research and Practice*, vol. 8, B. Antia, Ed., Amsterdam: John Benjamins, 2007, pp. 49–58. doi: doi:10.1075/tlrp.8.07sch.

[7] S. Abufardeh and K. Magel, "Software localization: The challenging aspects of Arabic to the localization process (Arabization)," in *Proceedings of the IASTED International Conference on Software Engineering*, SE 2008, in Proceedings of the IASTED International Conference on Software Engineering, SE 2008. Dec. 2008, pp. 275–279.

[8] P. Sandrini, "Localization and translation," in *LSP translation scenarios*, H. Gerzymisch-Arbogast, G. Budin, and G. Hofer, Eds., Vienna: The ATRC Group, 2008, pp. 167–191.

[9] M. Dingfelder Stone, "Authenticity, Autonomy, and Automation: Training Conference Interpreters," in *Towards Authentic Experiential Learning in Translator Education*, D. Kiraly, Ed., Göttingen: V&R unipress, 2015, pp. 113–128.

[10] D. Kiraly, "Towards a View of Translator Competence as an Emergent Phenomenon: Thinking Outside the Box(es) in Translator Education," in *New Prospects and Perspectives for Educating Language Mediators*, D. Kiraly, S. Hansen-Schirra, and K. Maksymski, Eds., Tübingen: Narr Verlag, 2013, pp. 197–224.

[11] D. Kiraly, "Beyond the Static Competence Impasse in Translator Education," in *Translation and Meaning*, M. Thelen, G.-W. van Egdom, D. Verbeeck, Ł. Bogucki, and B. Lewandowska-Tomaszczyk, Eds., Frankfurt: Peter Lang, 2016, pp. 129–142. [Online]. Available: https://api.semanticscholar.org/CorpusID:114932469

[12] P. Williamson, "The Translating Facilitator: an Empowerment Approach," in *Teaching Translation vs. Training Translators*, vol. 8, M. Kubánek, O. Klabal, and O. Molnár, Eds., Olomouc: Palacký University Olomouc, 2022, pp. 21–32.

[13] L. A. Leiva and V. Alabau, "The Impact of Visual Contextualization on UI Localization," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, in CHI '14. New York: Association for Computing Machinery, 2014, pp. 3739–3742. doi: doi:10.1145/2556288.2556982.

[14] J. C. Richards, *Communicative language teaching today*. Cambridge: Cambridge University Press, 2006.

[15] A. Kocbek, "Unlocking the potential of translation for FLT," *linguistica*, vol. 54, no. 1, pp. 425–438, Dec. 2014, doi: doi:10.4312/linguistica.54.1.425-438.

[16] D. Folaron, "A discipline coming of age in the digital age," in *Perspectives on Localization*, K. J. Dunn, Ed., Amsterdam-Philadelphia: John Benjamins Publishing, 2006, pp. 195–219.

[17] H. Kowalewski, *Pyrates*. (2023). Python. [Online]. Available: https://sourceforge.net/projects/pyrates-game/

[18] M. Karwatowska and J. Szpyra-Kozłowska, *Lingwistyka płci: ona i on w języku polskim*. Lublin: Wydawnictwo Uniwersytetu Marii Curie-Skłodowskiej, 2005.

[19] R. Pacheco Aguilar, "The Question of Authenticity in Translator Education from the Perspective of Educational Philosophy," in *Towards Authentic Experiential Learning in Translator Education*, D. Kiraly, Ed., Göttingen: V&R unipress, 2016, pp. 13–31.

[20] M. Polanyi, *The tacit dimension*. Chicago-London: University of Chicago Press, 2009.

[21] E. Veveris, "Continuous localization & translation: what is it & how to do it," Lokalise Blog. Accessed: Dec. 22, 2022. [Online]. Available: https://lokalise.com/blog/continuous-localization-101/

# Beyond Stability: Exploring Efficiency and Proficiency in Stable Marriage Problem with Hungarian Algorithm

Mateusz Oćwieja
Anna Sasak-Okoń*

## 1 Introduction and related work

Matching algorithms are algorithms used to solve graph matching problems in graph theory. These problems arise when there is a need to create a set of edges that do not share common vertices. These algorithms find applications in various fields, from resource allocation to network optimization. In practice, they involve connecting vertices in a graph using edges that do not have common vertices. For example, they can be used to form pairs of students in a class based on their qualifications or to establish pairs in a bipartite graph, where two sets of vertices must be connected. Such algorithms can find the maximum flow in a network, which can be then used to optimize transportation systems or network communication or solve assignment problems, like assigning workers to tasks in a way that minimizes the overall cost or maximizes operational efficiency [11].

The Stable Marriage Problem (SMP) is a classic problem in the field of matching theory and algorithms. It models the scenario of pairing an equal number of men and women based on their preferences for potential partners. The goal is to create stable marriages where there are no pairs of individuals who would both prefer to be with each other rather than with their current partners [5].

GSA has a wide variety of practical applications of the SMP. One of them is the assignment of graduating medical students to their first hospital appointments, found in The National Resident Matching Program (NRMP) [12]. The other common scenarios are matching primary school students to secondary schools, organs to patients, and even market trading [18, 8].

*Corresponding author — anna.sasak-okon@mail.umcs.pl

## 1.1 The Gale-Shapley Stable Marriage Algorithm

The Gale-Shapley Algorithm (GSA) stands out as a prominent solution in the field of matchmaking algorithms. Developed by mathematicians David Gale and Lloyd Shapley in the 1960s, this algorithm addresses the Stable Marriage Problem (SMP), where individuals on two sides each have preferences and the goal is to find a stable matching. The key feature of the GSA is its ability to guarantee a stable matching, meaning there are no pairs of individuals who would both prefer each other over their current partners. This stability ensures the absence of incentives for individuals to break their current matches in favor of others. It has been well described in College Admissions and the Stability of Marriage paper [6].

The GSA algorithm operates on the principle of proposing and accepting/rejecting proposals iteratively. Initially, individuals on one side propose to those on the other side based on their preferences. The receiving individuals tentatively accept the proposals from their most preferred suitors. If an individual receives multiple proposals, they choose the most preferred one and reject the others. This process continues until each individual is either matched with their most preferred partner or has proposed to all potential partners. After completing $O(n^2)$ steps, this algorithm generates a set of marriages that is both stable and optimal based on the preferences of the proposing side.

The further development of the algorithm was contributed by Alvin Roth, who was awarded the Nobel Prize in 2012 "for the theory of stable allocations and the practice of market design". Roth's extensive analysis has had a notable impact on both theoretical and practical aspects. He has demonstrated the algorithm's adaptability and relevance in real-world scenarios, such as assigning doctors to hospitals, students to schools, and allocating human organs for transplant [13].

Despite the fact that the issue of stable marriage affects a specific family of problems with numerous possibilities for extensions or modifications (e.g., the College Admission Problem, Rural Hospital Theorem), most of them can be reduced to the basic version of the problem and solved using the fundamental algorithm proposed by Gale and Shapley in [6], where the authors assert that these problems are analogous and essentially pertain to the same underlying problem.

Some researchers, like Teo in 2001, have looked into issues with the GSA algorithm. They've studied strategic problems in the Gale-Shapley stable marriage model, highlighting possible weaknesses. While the algorithm is good at finding stable solutions, scholars, including Teo, have questioned whether it can be easily manipulated. In certain situations, the algorithm's default preferences might be taken advantage of, challenging how strong its results are. More research, like Huang's work in 2005, explores strategy problems in the stable marriage issue under the GSA, giving more insights into potential problems. Consequently, a valid discussion on the robustness of the GSA exists, as authors report numerous concerns about strategic behaviors and the stability of the presented solutions [16, 7].

The GSA exhibits a slight issue with its default preferences, displaying a tendency towards favoritism. Despite its proficiency in finding stable solutions, the quality of these solutions may be constrained. Minor adjustments in individual preferences or weights within the algorithm can lead to significant changes in the final matches. This sensitivity to minor changes poses both a challenge and an opportunity. On one hand, it introduces unpredictability and raises questions about

the robustness of the algorithm. On the other hand, this sensitivity can be leveraged as an advantage. By strategically modifying preferences or weights, it becomes possible to explore different and potentially improved solutions to achieve more favorable matches. Some methods have been proposed to address this issue. For example, describing the stable matching polytope using linear inequalities involves refining these inequalities, eliminating redundant constraints, and adjusting preference lists to enhance the stability of matchings. This process, as outlined in [1], results in both firm-optimal and worker-optimal stable matches.

## 1.2 The Hungarian Algorithm

The Hungarian Algorithm (HA), developed by Dénes Kőnig and Jenő Egerváry in the 1930s, is a versatile method designed to address optimization and assignment problems. Focused primarily on the assignment problem, it aims to find the most efficient allocation of resources to tasks or individuals to jobs.

Operating on augmenting paths within a weighted bipartite graph, the algorithm minimizes the total cost or maximizes the total profit associated with the assigned tasks. This adaptability allows its application in various domains, including logistics and scheduling.

In educational institutions, the Hungarian algorithm proves valuable for Course Scheduling by efficiently assigning courses to instructors and classrooms, taking into account constraints such as room availability and instructor preferences [15]. Additionally, in the optimization of transportation routes, the Hungarian algorithm plays a crucial role. By assigning tasks, such as deliveries, to available resources like vehicles and drivers, it effectively minimizes the overall cost of delivering goods [17].

In 1957, James Munkres conducted a review of the algorithm and noted its strong polynomial nature [10]. Consequently, the algorithm has also become recognized as the Kuhn–Munkres algorithm. The extended version of the Kuhn-Munkres algorithm was successfully applied to the more complex problem of the Sailor Assignment Problem by Dasgupta et al. [3]. They utilized an extension of this algorithm for rectangular matrices introduced by Bourgeois and Lassalle in 1971 [2], enabling the algorithm to operate in situations where the numbers of agents and tasks are unequal. This means that the use of the Hungarian Algorithm is not limited to the classic form of the SMP.

Significant advancements in the time complexity of the Hungarian algorithm were achieved through notable modifications by Edmonds, Karp, and Tomizawa, as outlined in their seminal work on network flow problems [4]. Originally burdened with a time complexity of $O(n^4)$, their meticulous refinements targeted the augmenting path search process, introducing more efficient techniques for path identification and augmentation within the weighted bipartite graph. Simultaneously, they introduced improvements in data structures, optimizing key operations to streamline the algorithm's execution. As a result of these comprehensive enhancements, the algorithm's time complexity was successfully reduced from $O(n^4)$ to a more efficient $O(n^3)$.

The HA's computational complexity of $O(n^3)$ renders it impractical for large-sized problems. However, ongoing efforts to enhance its efficiency have resulted

in the development of more efficient versions of described algorithms. Notably, a MATLAB implementation has been introduced as a potential solution to address scalability concerns [14]. Moreover, there are methods that reduce the execution time of the Hungarian algorithm without affecting its computational complexity class. The paper [19] shows how a minor modification to the Hungarian algorithm for the linear assignment problem can drastically reduce execution times by up to 90%.

The HA finds application in the SMP, aiming to establish a perfect matching in a weighted bipartite graph. This involves creating a set of edges where each vertex is connected to exactly one edge. While the GSA is specifically tailored for SMPs, ensuring stability by eliminating blocking pairs, the HA does not inherently prioritize stability. To adapt the HA for stable matching, preferences lists from the GSA can be incorporated by assigning edge weights based on these preferences. However, this adaptation may not guarantee stability, as the HA primarily optimizes for weight rather than stability criteria. The manipulation of preferences, such as assigning higher costs to women's preferences, can influence the obtained outcomes/results [9].

While previous research has applied the HA to optimize cost and efficiency in various applications, these studies did not address the stability requirement critical to the SMP. In this work, the authors adapt the HA specifically for SMP by incorporating stability constraints and enhancing participant satisfaction, presenting a novel approach that combines both aspects.

## 1.3 Contribution

The idea of applying an algorithm other than GSA to SMP arose during work on a related problem called the College Admission Problem. The analysis of results provided by GSA clearly showed that while the matchings were stable, there was room for improvement in the overall satisfaction of participants with these matchings. To illustrate the problem with GSA, let's analyze a simple example:

For the problem size $N = 5$ (meaning that 5 men and 5 women should be matched based on their preference lists), let's consider two stable pairs from the final matching, $(m_1, w_1)$ and $(m_2, w_2)$, which resulted from proposals by men $m_1$ and $m_2$ as their $1^{\text{st}}$ choices. However, their partners $w_1$ and $w_2$ treat them as their least favorable options, assigning men to them at their last $5^{\text{th}}$ position on their preference lists. The IPSS[1] for these two pairs is $(2, 10)$. If it turns out that $m_1$ and $m_2$ have, respectively, $w_2$ and $w_1$ as their $2^{\text{nd}}$ choices, and women $w_1$ and $w_2$ have men $m_2$ and $m_1$ as their $1^{\text{st}}$ choices, then the matching of pairs $(m_1, w_2)$, $(m_2, w_1)$ would not disturb the stability of the entire matching but would reduce the IPSS to $(4, 2)$. Thus, there is potential to maintain the stability of this solution with a possible significant reduction in IPS[2]) .

This challenge remained unaddressed with GSA, prompting the exploration of alternative algorithms. The quest was for an algorithm that, in ensuring stability, would also prioritize the quality of matchings. To achieve this, the chosen algorithm

[1] In the context of this example, for improved clarity, the IPSS treats the $1^{\text{st}}$ preference as having an index of 1.

[2] Inversed Preference Score (IPS), described in Section 2.3

needed to globally consider preferences, in contrast to the local approach of GSA. Following a thorough analysis of available options, HA emerged as the algorithm most closely aligned with SMP.

This work concentrates on modifying the HA to address the SMP more efficiently. Consequently, utilizing the supplied data that depicts the number of participants (men and women) along with lists of their preferences, the goal is to produce a stable matching using the HA. Nevertheless, two fundamental problems were encountered.

Firstly, the HA cannot function directly with preference lists. It necessitates the representation in the form of a complete weighted bipartite graph. This problem was overcome by introducing marriage formation cost and creating a common cost matrix, as elaborated in Section 2.2. For the purpose of calculating the marriage formation cost, formulas for symmetric and asymmetric transformations were developed (defined in Table 1), enabling diverse interpretation of values recorded in preference lists, especially for selecting any degree of favoritism towards the parties.

Secondly, the HA lacks the capability to ensure the stability of a matching, as it inherently addresses the SMP in an approximate manner. In response to this issue and to guarantee the stability of the solution, a supplementary algorithm was developed and implemented. This stabilizing algorithm operates on an initially unstable match, examining all unstable pairs. Through successive iterations, it strategically exchanges unstable partners in a greedy approach, progressively reducing the number of unstable pairs until their complete elimination, as explained in more detail in Section 2.4.

To determine whether HA is capable of solving SMP and to compare its results with GSA, a set of parameters has been defined in Section 2.3. The purpose of these parameters is to measure the quality of the final matching. The quality of the matching can be assessed by addressing the following questions:

- Is the matching stable?
- If the matching is unstable, how many unstable pairs does it contain?
- What is the general satisfaction of participants? (Measured as the sum of preference priorities used in the matching)
- How significant is the disproportion of satisfaction between men and women? (Defined by degree of favoritism)

Matching is considered of high quality if it is stable (or has a minimum number of unstable pairs), and the general satisfaction is relatively high. The degree of favoritism can influence the quality in different ways, depending on the scenario, but generally, we would prefer to observe balanced satisfaction, without preferential treatment.

To compare both algorithms, a series of tests was conducted using randomly generated data. The experimental data consisted solely of preference lists generated with a uniform distribution. For the defined problem of size N, a random permutation of length N was generated for each of the N men and N women. The results presented in Section 3 represent averaged outcomes for selected representative test samples in each case.

In Section 3, a comparison of both algorithms is provided, along with their characteristics based on the defined measurement parameters. The results indicate that HA offers greater flexibility in balancing IPSS, and strategically chosen transformations outperform GSA in terms of matching quality. Interestingly, prioritizing DoF negatively affects IPS, suggesting a compromise between the two aspects.

For the results obtained by HA, further processing was conducted using a greedy stabilizing algorithm, and the observational outcomes are presented in Section 3.3. The algorithm demonstrates high effectiveness in rectifying unstable pairs and reducing IPS. A strong correlation is observed between algorithm iterations and participants' satisfaction gain. Each unstable matching is successfully resolved, affirming the validity of the greedy algorithm concept.

# 2 Selected implementation details

In our implementation, we decided to explore the behavior of the GSA and the HA algorithms concerning the foundational problem. We examine the performance of selected algorithms for the basic SMP, where the number of men and women and their preferences are equal. For a problem of size $N$, it is possible to define a preference matrix with dimensions $N \times N$. The result of the algorithm's operation on this matrix should be a one-to-one function, assigning $N$ men to $N$ women (resulting in a perfect matching).

## 2.1 Encoding

Both GSA and HA take N as an input value, representing the number of vertices on each side of a symmetric bipartite graph, along with the preferences of men and women, which define the edges of the graph. Each man and woman has their own preference list of length N, with preferences encoded using integers. The preference list stores them in the form of a permutation of numbers with values ranging from 0 to N-1. In such a list, each value corresponds to a person of the opposite sex, and its position defines the priority, meaning at index 0, there is always the most preferable partner. For example, if the preference list for man $m_0$ looks like $[1, 2, 0]$, it signifies his preferences for women as follows: $w_1, w_2, w_0$ (from most to least desirable). Regarding the output data, the index of the list represents a specific man, and the value at the index represents the woman assigned to him (forming a pair of connected vertices in the final graph). If the result of the above example is the following vector $[2, 0, 1]$, it means the final pairs are: $(m_0, w_2), (m_1, w_0), (m_2, w_1)$.

Before being passed to the algorithm, the received preference lists are transformed into two preference arrays/matrices for the GSA algorithm (separate for men and women), or a common cost matrix for the HA algorithm, containing combined information from both arrays.

## 2.2 Applying Hungarian Algorithm to Stable Marriage Problem

Considering that the Hungarian Algorithm (HA) operates on a complete weighted bipartite graph, the edge costs can be determined by assigning each edge

a "marriage formation cost." This cost can be calculated using various formulas, such as adding values corresponding to their priority preferences associated with a given pair.

In assessing the marriage formation cost, we apply the argument sort function to preference lists. This function essentially provides the indices that would arrange an array in ascending order, thus indicating the significance of each woman to a man within the context of pair formation. For instance, if the original preferences were $[3, 1, 0, 2]$, the argument sort function would yield $[2, 1, 3, 0]$. This transformation enables us to view preferences from a new perspective. Now, the index in the array represents a partner of the opposite sex, while the corresponding value denotes the priority (with lower values indicating higher priority).

$$\begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{bmatrix} \circ \begin{bmatrix} w_{11} & w_{12} & w_{13} \\ w_{21} & w_{22} & w_{23} \\ w_{31} & w_{32} & w_{33} \end{bmatrix}^T = \\ = \begin{bmatrix} m_{11} \circ w_{11} & m_{12} \circ w_{21} & m_{13} \circ w_{31} \\ m_{21} \circ w_{12} & m_{22} \circ w_{22} & m_{23} \circ w_{32} \\ m_{31} \circ w_{13} & m_{32} \circ w_{23} & m_{33} \circ w_{33} \end{bmatrix} \tag{1}$$

Formula 1 shows how two preference matrices for men (left matrix) and woman (right matrix), that were transformed by argument sort, are joined to obtain a common cost matrix. We need to apply a transformation to one of the matrices so that the common matrix represents a bipartite graph with edges between men and women (in this case, men represented by rows and women by columns). To obtain the weights, we perform a certain operation between two variables, for example, addition.

The resulting matrix, representing the weighted bipartite graph, can be directly utilized by the HA to obtain a perfect matching. Based on applied operation, seeking the optimal cost (minimum) may yield a matching resembling stability.

$$\begin{bmatrix} 0 & 1 & 2 \\ 2 & 0 & 1 \\ 1 & 0 & 2 \end{bmatrix}^A \circ \begin{bmatrix} 1 & 2 & 0 \\ 2 & 1 & 0 \\ 2 & 0 & 1 \end{bmatrix}^{AT} = \begin{bmatrix} 0 & 1 & 2 \\ 1 & 2 & 0 \\ 1 & 0 & 2 \end{bmatrix} \circ \begin{bmatrix} 2 & 0 & 1 \\ 2 & 1 & 0 \\ 1 & 2 & 0 \end{bmatrix}^T = \\ = \begin{bmatrix} 2 & 6 & 9 \\ 9 & 2 & 2 \\ 2 & 1 & 6 \end{bmatrix} \tag{2}$$

$$(p_1 \circ p_2) = (p_1 + 1) \cdot (p_2 + 1) \tag{3}$$

Example represented by Formula 2 shows how the common cost matrix can be obtained from lists of preferences. At first, the argument sort function is applied to each matrix (denoted by A), storing preferences lists. Then, the operation $\circ$, defined as multiplication of incremented preferences (3) is applied. We implement incrementation prior to multiplication in the operation $\circ$ to differentiate between scenarios where both partners or only one partner is matched with their first choice (multiplying by 0 would result in the loss of relevant information). The presented

transformation process treats preferences on both sides symmetrically, suggesting that finding the global minimum of the cost function will result in a relatively balanced outcome without favoring either side. Nevertheless, the results for this set of preferences, obtained from both GSA and HA are the same. The result matching is: $[(m_0, w_0), (m_1, w_2), (m_2, w_1)]$, which is men-optimal, since each man got paired with his first choice partner.

## 2.3 Measurement parameters and implemented cost transformations

To allow proper comparison of the algorithms, some additional methods were developed. The most important factor, in terms of matchings produced by HA, was measurement of stability, since the HA cannot guarantee a stable matching. It was achieved by verifying the absence of two pairs $m, w$ and $m', w'$, where both individuals prefer each other's partners $m, w'$ and $m', w$.

The other defined parameter was the number of unstable pairs (UPC), which was determined as the count of potential changes in pairs, where one pair could be counted multiple times if it could be matched with several different pairs.

The next important parameter was the Inversed Preference Score by Sex (IPSS), represented by a tuple $(\mathrm{IPS}_M, \mathrm{IPS}_W)$. It stored the sum of preference priorities used in the final matching for both sides separately. A lower value of IPSS indicates that the final matching was achieved by pairing more-preferable partners with their corresponding side. Similarly, the Inversed Preference Score (IPS) represents the general sum of IPSS for both sexes. The lower the score, the better

The measure of favoritism, determined by the IPSS balance between two values $m$ and $w$ (representing IPSS for men and IPSS for women respectively), and yielding a signed result, can be computed using a metric denoted as $\mathrm{DoF}(m, w)$. The formula 4 for this metric is expressed as:

$$\mathrm{DoF}(m, w) = \frac{m - w}{\max(m, w)} \tag{4}$$

DoF returns values in range of $[-1, 1]$, where 0 signifies a perfect balance (lack of favoritism). Higher values of DoF indicate higher favoritism of men's preferences.

The Hungarian algorithm works with edge weights in a bipartite graph, and these weights can be determined for one set of preferences in various ways. In our investigation, we categorized methods for calculating these weights into two groups: symmetric and asymmetric.

Symmetric transformations are those that, while manipulating preference priorities, treat both sides of the graph equally and impartially. This implies that neither men nor women should receive preferential treatment through these methods. We achieve this by symmetrically handling the preferences of each side.

Conversely, there are asymmetric transformations that introduce a bias. For example, one may employ a weight system where men's preferences are multiplied by a specific constant. With this approach, setting the weights to zero allows for complete disregard of one side, providing flexibility based on the scenario. It is important to note that such asymmetric transformations enable the introduction

Table 1: Formulas for Symmetric and Asymmetric Transformations in Edge Weight Calculations for Hungarian Algorithm

| Type | Name | Formula |
|---|---|---|
| Symmetric | mult | $(m+1)\cdot(w+1)$ |
| | squared | $(m+w)^2$ |
| | plus | $m+w$ |
| | +pow | $(m+w)^{\mathrm{x}}$ |
| | pow+ | $m^{\mathrm{x}}+w^{\mathrm{x}}$ |
| Asymmetric | only-men | $m$ |
| | fav-men | $(m+1)^2\cdot(w+1)$ |
| | fav-men* | $(m+1)\cdot\mathrm{x}+w$ |
| | fav-men^ | $(m+1)^{\mathrm{x}}+w$ |

of favoritism, as the weighted values can be adjusted to give preference to a specific group.

Table 1 defines transformation functions, that operate on pairs of preference priorities $(m, w)$, used in this research. They use simple mathematical operations, like addition, multiplication and exponentiation. There is a possibility of further expanding these transformations and introducing formulas; however, in this study, we decided to limit ourselves only to these basic formulas. For better clarity, among the asymmetric transformations, only those related to men are presented. For women, they are analogous.

## 2.4 Greedy algorithm responsible for stabilizing marriage

Since the Hungarian method cannot guarantee a stable matching, an attempt was made to develop a method to improve the stability of a specific matching. While it is known that a stable matching always exists [6], if we encounter an unstable matching as an output from the algorithm (HA), we can examine the unstable pairs within that matching and attempt to stabilize it by addressing the unstable pairs. A pair is considered unstable when men (or women) express a preference to exchange their current partners, leading to the dissolution of two existing marriages and the formation of new ones. The objective is to create new pairings where everyone involved is at least as satisfied as they were in the previous marriages. It is done by simply finding unstable pairs and swapping their partners.

To enhance the effectiveness of this method, prior to each swap, we can optimize the selection process by identifying two pairs with the highest potential for IPS reduction. This potential is defined as the difference between the IPS of the matching after and before the swap. Opting for pairs with the greatest potential IPS reduction ensures a significant decrease in partner priorities, allowing individuals to secure the most preferable partners available among all unstable pairs. This minor adjustment in the selection process consistently leads to choosing a local optimum, thereby reducing the number of iterations required to stabilize the matching and improving its overall quality (in terms of IPS and UPC).

Listing 1: Pseudo code for a Greedy Algorithm to Stabilize Marriages

```
Function stabilizeMatching(initialMatching, preferences):
    currentMatching = initialMatching
    unstablePairs = findUnstablePairsWithPotentialIPS(
                        initialMatching, preferences)
    while unstablePairs is not empty:
        bestImprovement = sortPairsByImprovementScore(
                            unstablePairs).top()
        updateMatching(currentMatching, bestImprovement)
        unstablePairs = findUnstablePairs(
                          currentMatching, preferences)
    return currentMatching
```

The stabilizing algorithm we propose, as outlined in Listing 1, follows specific steps. During each iteration, it identifies unstable pairs in the current matching and selects the two most promising pairs in terms of potential IPS reduction. Subsequently, it swaps partners between these two pairs, updating the current matching. This process iterates until all unstable pairs are resolved. In this research, the algorithm successfully addressed all detected unstable matchings.

$$\begin{bmatrix} (0,4) & (2,3) & (1,1) & (3,4) & (4,3) \\ (2,1) & (0,0) & (1,4) & (4,0) & (3,0) \\ (1,0) & (0,2) & (2,2) & (4,3) & (3,4) \\ (0,3) & (4,4) & (2,3) & (3,2) & (1,2) \\ (0,2) & (3,1) & (1,0) & (2,1) & (4,1) \end{bmatrix} \tag{5}$$

For better problem understanding, the following example of size $N = 5$ can be considered. For the preference matrix presented above (Formula 5), there is an unstable matching $(3, 0, 2, 1, 4)$, containing set of 4 unstable pairs:

$$(0,3) \text{ with } (3,1), \quad (1,0) \text{ with } (3,1), \quad (2,2) \text{ with } (3,1), \quad (3,1) \text{ with } (4,4)$$

We can apply a greedy algorithm for stabilizing the marriage: At first, let's try using the version of the algorithm that always selects the first unstable pair found. We perform 1st swap: $(0,3)$ and $(3,1)$ into $(0,1)$ and $(3,3)$. Now unstable pairs are:

$$(0,1) \text{ with } (2,2), \qquad (3,3) \text{ with } (4,4)$$

Then, We perform 2nd swap: $(0,1)$ and $(2,2)$ into $(0,2)$ and $(2,1)$. Now unstable pairs are:

$$(3,3) \text{ with } (4,4)$$

Finally, We perform 3rd swap: $(3,3)$ and $(4,4)$ into $(3,4)$ and $(4,3)$. After this swap, the matching becomes stable. The matching is stabilised after 3 iterations, resulting in the final output: $[2, 0, 1, 4, 3]$.

Now. let's use an enhanced version of the algorithm that selects the most promising pairs: For the same unstable matching $(3, 0, 2, 1, 4)$, containing 4 unstable pairs, it would find the most promising pairs as $(1, 0)$ and $(3, 1)$ and would perform just one swap into new pairs: $(1, 1)$ and $(3, 0)$. After this swap, the matching becomes stable. **We can see that the proposed algorithm stabilized matching after 1 iteration**, resulting in the final output: $[3, 1, 2, 0, 4]$.

In this simple example, it can be observed that not only did the enhanced algorithm version require fewer iterations to stabilize the matching, but it also returned

Table 2: Time complexity and IPS comparison by N

| Size (N) | Time (seconds) | | Quality (IPS) | |
|---|---|---|---|---|
| | GSA | HA | GSA | HA |
| 3 | $4.94 \times 10^{-5}$ | $1.33 \times 10^{-3}$ | 3.72 | 3.4 |
| 5 | $5.57 \times 10^{-5}$ | $1.48 \times 10^{-3}$ | 11.12 | 9.75 |
| 10 | $1.2 \times 10^{-4}$ | $4.29 \times 10^{-3}$ | 40.44 | 35.15 |
| 30 | $5.6 \times 10^{-4}$ | $8.984 \times 10^{-2}$ | 291.2 | 223.0 |
| 50 | $1.42 \times 10^{-3}$ | 0.4188 | 702.16 | 512.05 |
| 100 | $6.29 \times 10^{-3}$ | 4.03 | 2416.72 | 1500.3 |
| 300 | $5.045 \times 10^{-2}$ | 146.2 | 16615.04 | 8053.65 |

a more similar matching to the original one. In contrast to the basic version, it did not produce any new unstable pairs during the process of stabilization.

# 3 Experimental results

The experiments were divided into three main parts. In Section 3.1, a comparison between GSA and HA was conducted, focusing on execution time and overall quality measured in IPS. Additionally, an analysis of the DoF was performed by comparing GSA to HA using multiple transformations.

Next, in Section 3.2, we evaluated the effectiveness of different HA transformations, as outlined in Table 1. In our evaluation, we emphasized the importance of determining the optimal transformation. We compared their characteristics and checked the trade-offs involved in balancing IPS, UPC, and DoF.

Finally, in Section 3.3, we conduct tests on selected transformations and identify unstable matchings. We then subject these unstable matchings to the greedy algorithm for stabilizing marriages (presented in Listing 1).

## 3.1 GSA vs HA: complexity and quality of the matching

Table 2 illustrates a comparison of execution time and matching quality between GSA and HA. The results from time measurements conducted on the smallest tests ($4.94 \times 10^{-5}$ s vs $1.33 \times 10^{-3}$ s) suggest that even for smaller tests, HA requires more time, hinting at a potentially larger constant component and overhead. The time difference becomes more pronounced with the test size: for N=3, HA executed about 27 times slower, and for N=300, it was almost 2.9 thousand times slower. This corresponds to the theoretical complexities of these algorithms being $O(n^2)$ and $O(n^3)$, respectively. Also, it's important to mention that the execution time measurements of HA include the time required to transform the shared preference matrix into the shared cost matrix (i.e., using transformation). This transformation took 1.12s for the largest test (N=300) and made up 0.77% of the total execution time.

At this juncture, one might conclude that due to significantly higher time complexity, using HA might not be worthwhile, given that GSA is at least an order of magnitude faster for larger tests. However, execution time shouldn't be the sole cri-

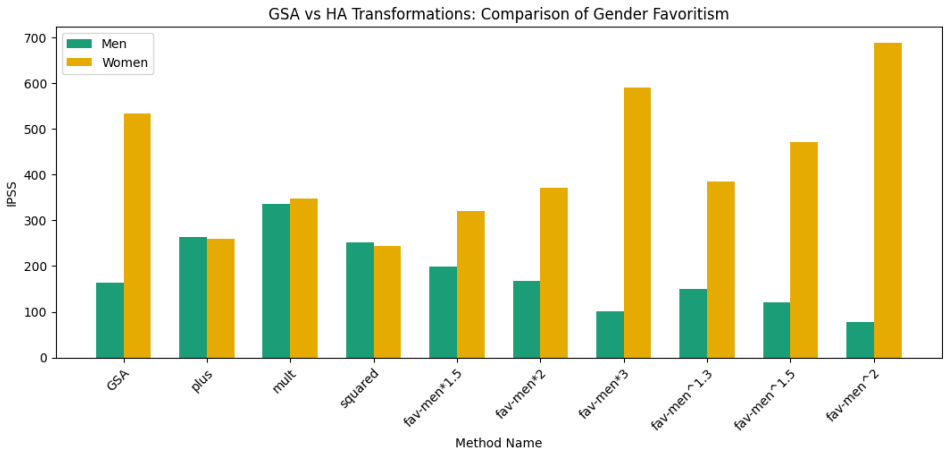



Figure 1: Comparison of GSA and different HA transformations in terms of gender favoritism

terion for evaluating algorithms and the quality of the results they provide should also be considered.

This quality, measured in terms of IPS, is detailed in two right columns (labeled Quality) of Table 2. In all examined cases, HA yielded better quality measurement results, consistently obtaining lower IPS values. For N=3, HA achieved a result 1.09 times better, and for N=300, a significantly **better result of 2.06 times**. This difference is substantial and grows with the participant pool's size. The figures for N=300 indicate that, on average, **participants received a partner ranked at position 28 (GSA) vs 13 (HA)**. This implies that despite the longer execution time, HA provides results that significantly better satisfy the participants and should be considered as an alternative method for solving the SMP.

To ensure a comprehensive quality comparison between GSA and HA, it should additionally consider the stability degree of HA and the degree of bias in relation to GSA. To address this, a test was conducted with a fixed number of participants N=50, spanning 100 random samples. The observed results are showcased in Fig. 1 and Table 3.

Based on the algorithm's characteristics, GSA exhibited a preference for men over women with a DoF of 0.69 and IPS of 697. This should serve as a benchmark for interpreting other results. Symmetrical transformations were applied to achieve a comparable level of balancing preferences on both sides, fluctuating around perfect equilibrium. The resulting DoF values for each transformation were as follows: 'plus': -0.02, 'mult': 0.03, 'squared': -0.03. Among these, the 'squared' transformation stood out with the best IPS result, reaching 496. This was **40.5% superior to GSA**, 37.7% better than 'mult,' and 5.2% more effective than 'plus.' In conclusion, the three symmetric transformations successfully maintained balance, and the 'squared' transformation, with the lowest IPS, emerged as the most effective among them. Furthermore, unlike 'plus' and 'mult', the 'squared' transformation consistently produced stable matchings, as shown in Table 3."

Table 3: Number of Unstable Matchings for Functions in Fig. 1

| | |
|---|---|
| GSA | 0 |
| plus | 7 |
| mult | 1 |
| squared | 0 |
| fav-men*1.5 | 2 |
| fav-men*2 | 4 |
| fav-men*3 | 5 |
| fav-men^1.3 | 0 |
| fav-men^1.5 | 1 |
| fav-men^2 | 2 |

When comparing asymmetric transformations, as shown in Fig. 1, it becomes evident that elevating favoritism toward one side results in a disproportionate increase for the other side, consequently leading to a rise in overall effectiveness. Specifically, the augmentation of the X coefficient, which amplifies bias toward one side's preferences, results in an escalated DoF and IPS. Examining the transformations with X employed as a multiplier, the successive changes in IPSS were as follows: $1.5 \rightarrow 2.0$: $(-15.8\%, 16.3\%)$, $2.0 \rightarrow 3.0$: $(-40.2\%, 58.7\%)$, and for transformations, using X as an exponent: $1.3 \rightarrow 1.5$: $(-19.7\%, 22.5\%)$, $1.5 \rightarrow 2.0$: $(-35.1\%, 46.0\%)$.

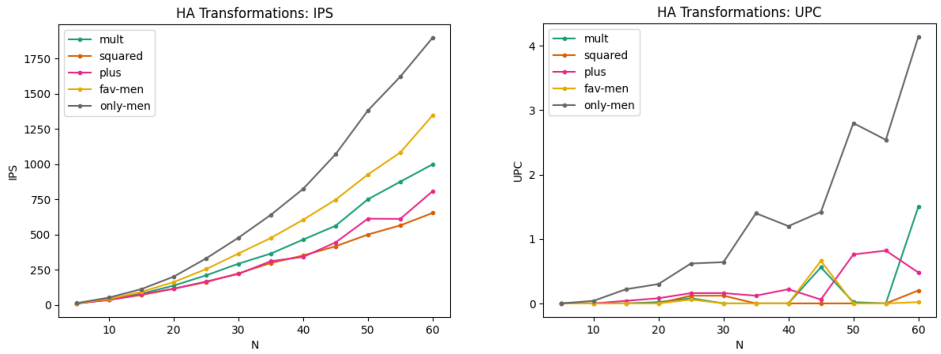
This observation implies that we have the latitude to select the degree of favoritism, and the algorithm will adjust accordingly. However, this adaptability comes at the expense of a reduction in overall satisfaction, as measured by IPS. Furthermore, as indicated in Table 3, an increase in the X coefficient correlates with a slight rise in unstable matchings affecting UPC.

Interestingly, the transformations labeled 'fav-men^1.3' and 'fav-men^1.5', employing exponents of 1.3 and 1.5, respectively, outperformed GSA, as shown in Fig. 1. Unexpectedly, however, these transformations resulted in lower IPSSs for both sexes. Additionally, the transformation with an exponent of 1.3 achieved UPC of 0, indicating its stability across all samples. This suggests the presence of a viable alternative to GSA, as asymmetric transformations applied to HA can uncover solutions with a high likelihood of stability while still favoring one side, demonstrating higher effectiveness as measured by both IPS and IPSS.

## 3.2 HA transformations: efficiency comparison

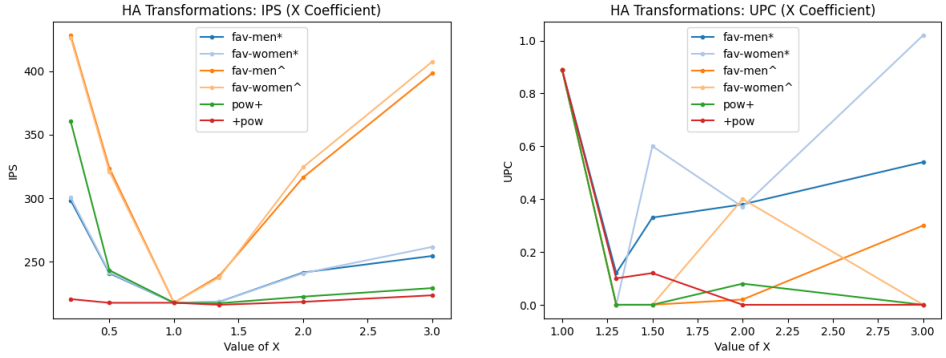
The effectiveness of matching generated by HA varies based on the applied transformation. The quality of results obtained by HA after applying different transformations, based on formulas outlined in 1, is illustrated in two sets of figures (Fig. 2 and Fig. 3). In both of them, it is crucial to regard the 'only-men' transformation (depicted by the grey line) as a benchmark, as it solely takes into account preferences on one side. Any outcome inferior to this transformation should be promptly dismissed. Nevertheless, as expected, the remaining transformations demonstrate favorable characteristics compared to this extreme approach.

The first set focuses on static transformations (independent of the parameter x)



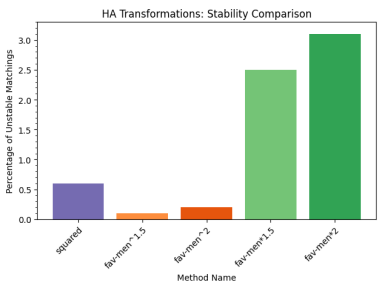

(a) Quality by IPS (b) Quality by UPC

Figure 2: Matching Quality Comparison of HA Over Transformations

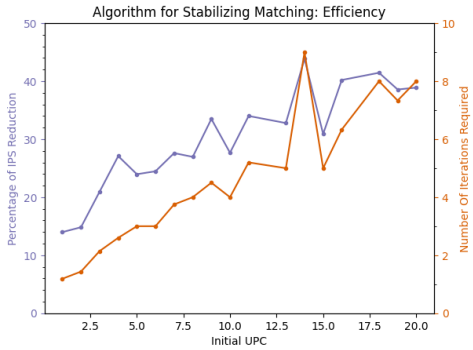



(a) Quality by IPS (b) Quality by UPC

Figure 3: Matching Quality Comparison of HA Over Transformations using X Coefficient




(a) Unstable Matchings Percentage produced by selected Hungarian Transformations




(b) Stabilizing Algorithm Efficiency

Figure 4: Unstable Matchings produces by HA and fix applied to them by Stabilizing Algorithm

and is illustrated in Fig. 2. Among static transformations, the 'squared' transformation stands out with the most favorable results. Based on the conducted tests, it can be inferred that 'squared' should be the default choice when emphasizing the preservation of the lowest IPS and high stability A comparison between two other symmetric methods, 'plus' and 'mult,' is presented in 2a. The analysis indicates that the 'plus' method exhibits slightly better characteristics in terms of IPS. However, referring to 2b, it becomes evident that the 'plus' method, for each test size, resulted in some unstable matchings. In contrast, the 'mult' method can be considered more stable, as it showed some instability only for larger N.

The second set involves dynamic transformations (with a variable parameter x) and is represented in Fig. 3. We can see the impact of X coefficient on IPS (Fig. 3a) and on UPC (Fig. 3b). All transformations meet at X=1, as they degenerate to the same function. Looking at this pair of plots the general conclusion may come upon: the increase of X coefficient has negative impact on IPS, but may decrease UPC. The link between X and IPS rise for assymetric transformations (observed on Fig. 3a), leads to conclusion that as the degree of favoritism towards one side increases, the IPS indicator deteriorates. Therefore, **the stronger the bias, the lower the overall satisfaction** with the match.

The comparison above, evaluating the performance of the HA across different transformations converting the preference matrix into a cost matrix, highlights the crucial role of selecting an appropriate transformation for a specific problem.

## 3.3 Greedy algorithm: solution to unstable matching

A conclusive test was conducted to demonstrate the effectiveness of the greedy algorithm in stabilizing marriages, as discussed in Section 2.4. The outcomes of this evaluation are depicted in Figure 4. During the test, which involved comparing 5 selected transformations (see Figure 4a) on 1,000 samples with N varying from 5 to 50, instances of unstable matchings were identified. Subsequently, these unstable matches were isolated and processed using the greedy algorithm. The results ob-

tained from running this algorithm are illustrated in the chart presented in Figure 4b.

Among the 5 selected transformations, these represented by exponential formulas yielded the most favorable results in terms of UPC, with 'fav-men-1.5' achieving an impressive reduction to only 0.1% unstable matches. Nevertheless, the only symmetric transformation performed commendably as well, securing the 3rd position with a modest 0.6% unstable matches. Three conclusions can be drawn from the observations in Figure 4a:

1. An increase in the $X$ coefficient correlates with a rise in the number of unstable matches.

2. Transformations utilizing exponential formulas exhibit greater stability than their linear counterparts.

3. The most stable transformations are asymmetric ones, utilizing $X$ values close to one as exponents.

Examining the chart in Fig. 4b, a robust **correlation of .96 between IPS gain and the number of iterations** (stabilized pairs count) is evident. This implies that the reduction of unstable pairs is closely linked to the improvement of IPS. In some instances, as many as 20 unstable pairs were identified, and they were stabilized after 8 iterations. The required number of iterations consistently remained lower than the count of unstable pairs, indicating the optimal exploitation of local minima by the greedy algorithm. This leads to the conclusion that the presented algorithm for stabilizing marriage is not only effective in reducing the number of unstable pairs and stabilizing matchings but also successfully diminishes overall IPS. This is a noteworthy feature, as it allows us to utilize the algorithm to enhance the quality of unstable matching without compromising its IPS and, in fact, significantly reducing it.

# 4 Conclusions

In this article, we applied the Hungarian Algorithm (HA) to the Stable Marriage Problem (SMP) and provided a detailed comparison with the Gale-Shapley Algorithm (GSA), the default algorithm for solving SMP. Both algorithms have proven effective in addressing the SMP, revealing distinct performance characteristics. Additionally, as HA doesn't guarantee stable matching, the new greedy algorithm for stabilizing marriage has been described and successfully tested among unstable marriages, proving it's effectiveness.

The GSA stands out for its unparalleled execution time, making it the preferred choice for large-scale problems. However, it lags behind the HA in terms of the final matching's quality. This improvement in matching quality comes at the expense of an evaluation time significantly slower than GSA. This emphasizes the nuanced trade-off between execution time and matching quality, necessitating careful consideration when selecting between the GSA and the HA for specific applications.

The notable feature of HA is its flexibility, enabling the free selection of cost transformations that best suit our scenario. This flexibility allows for balancing

IPSS by symmetrically and equitably treating both men's and women's preferences. When employing asymmetric transformations, any Degree of Favoritism (DoF) can be achieved, but it comes at the expense of compromising overall satisfaction among participants. Furthermore, the flexibility of HA allows for precise parameter adjustments, extending beyond preference priorities, which may accommodate a broader range of applications compared to what GSA can handle.

During this research, HA demonstrated greater effectiveness by generating higher-quality matchings while maintaining lower IPS compared to GSA. The most promising transformations were found to be 'squared' and 'fav-men^1.3', exhibiting relatively low UPC and a significant reduction in IPS (over 2 times better than GSA for medium-size tests). Interestingly, the asymmetric transformation 'fav-men^1.3' as the only one yielded the matching with lower IPSS than GSA for both sexes. This transformation, in combination with HA, demonstrated higher effectiveness in every aspect compared to the results obtained from GSA. This implies that certain cost transformations, when utilized in conjunction with HA, produce superior matchings, contributing to heightened participant satisfaction. Surprisingly, these improvements come without any downsides, except for an extended computational time.

The issue of instability in HA need not be a cause for concern, given the proposed algorithm specifically designed to stabilize matchings. It has proven highly effective, rectifying up to 20 unstable pairs in a mere 8 iterations. Furthermore, a substantial correlation of 0.96 between the number of resolved pairs and satisfaction gain was observed, signifying a noteworthy reduction in IPS.

In future research, it is valuable to use data derived from real-life scenarios. Furthermore, subsequent studies could concentrate on solving advanced variants of the SMP or even experiment with algorithms other than the HA.

# References

[1] Brian Aldershof, Olivia M Carducci, and David C Lorenc. Refined inequalities for stable marriage. *Constraints*, 4:281–292, 1999.

[2] Francois Bourgeois and Jean-Claude Lassalle. An extension of the munkres algorithm for the assignment problem to rectangular matrices. *Communications of the ACM*, 14(12):802–804, 1971.

[3] Dipankar Dasgupta, German Hernandez, Deon Garrett, Pavan Kalyan Vejandla, Aishwarya Kaushal, Ramjee Yerneni, and James Simien. A comparison of multiobjective evolutionary algorithms with informed initialization and kuhn-munkres algorithm for the sailor assignment problem. In *Proceedings of the 10th annual conference companion on Genetic and evolutionary computation*, pages 2129–2134, 2008.

[4] Jack Edmonds and Richard M Karp. Theoretical improvements in algorithmic efficiency for network flow problems. *Journal of the ACM (JACM)*, 19(2):248–264, 1972.

[5] Enrico Maria Fenoaltea, Izat B Baybusinov, Jianyang Zhao, Lei Zhou, and Yi-Cheng Zhang. The stable marriage problem: An interdisciplinary review from the physicist's perspective. *Physics Reports*, 917:1–79, 2021.

[6] David Gale and Lloyd S Shapley. College admissions and the stability of marriage. *The American Mathematical Monthly*, 69(1):9–15, 1962.

[7] Chien-Chung Huang. How hard is it to cheat in the gale-shapley stable matching algorithm. 2005.

[8] Kazuo Iwama and Shuichi Miyazaki. A survey of the stable marriage problem and its variants. In *International conference on informatics education and research for knowledge-circulating society (ICKS 2008)*, pages 131–136. IEEE, 2008.

[9] Duc Duong Lam, Van Tuan Nguyen, Manh Ha Le, Manh Tiem Nguyen, Quang Bang Nguyen, and Tran Su Le. Weighted stable matching algorithm as an approximated method for assignment problems. In *Asian Conference on Intelligent Information and Database Systems*, pages 174–185. Springer, 2020.

[10] James Munkres. Algorithms for the assignment and transportation problems. *Journal of the society for industrial and applied mathematics*, 5(1):32–38, 1957.

[11] Jing Ren, Feng Xia, Xiangtai Chen, Jiaying Liu, Mingliang Hou, Ahsan Shehzad, Nargiz Sultanova, and Xiangjie Kong. Matching algorithms: Fundamentals, applications and challenges. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 5(3):332–350, 2021.

[12] Jeremy R Rinard, Ben D Garol, Ashvin B Shenoy, and Raman C Mahabir. Successfully matching into surgical specialties: an analysis of national resident matching program data. *Journal of graduate medical education*, 2(3):316–321, 2010.

[13] Alvin E Roth. Deferred acceptance algorithms: History, theory, practice, and open questions. *international Journal of game Theory*, 36:537–569, 2008.

[14] Kartik Shah, Praveenkumar Reddy, and S Vairamuthu. Improvement in hungarian algorithm for assignment problem. In *Artificial Intelligence and Evolutionary Algorithms in Engineering Systems: Proceedings of ICAEES 2014, Volume 1*, pages 1–8. Springer, 2015.

[15] Shinsuke Tamura, Yuki Kodera, Shuji Taniguchi, and Tatsuro Yanase. Feasiblity of hungarian algorithm based scheduling. In *2010 IEEE International Conference on Systems, Man and Cybernetics*, pages 1185–1190. IEEE, 2010.

[16] Chung-Piaw Teo, Jay Sethuraman, and Wee-Peng Tan. Gale-shapley stable marriage problem revisited: Strategic issues and applications. *Management Science*, 47(9):1252–1267, 2001.

[17] Nobuaki Tomizawa. On some techniques useful for solution of transportation network problems. *Networks*, 1(2):173–194, 1971.

[18] Tarmo Veskioja. *Stable marriage problem and college admission.* TUT Press, 2005.

[19] MB Wright. Speeding up the hungarian algorithm. *Computers & Operations Research*, 17(1):95–96, 1990.

# Application of Temporal Convolutional Networks for Precise Detection of Artifacts in the EEG Signal Based on ICA Components

Anna Gajos-Balińska*
Bart Vanrumste
Grzegorz M. Wójcik

## 1 Introduction

### 1.1 Electroencephalographic signal

Electroencephalography (EEG) is a non-invasive method for monitoring the electrical activity of the brain, used in both scientific research and clinical diagnosis. This method is based on recording the fluctuations of electrical potentials generated by brain neurons, using electrodes placed on the scalp of the patient. EEG plays a key role in diagnosing and monitoring neurological disorders, such as epilepsy, sleep disorders, or brain damage [16, 15]. It is also a valuable tool in research on brain function, including in fields such as psychology, neuropsychology, and neuroinformatics.

One of the main challenges in electroencephalography is ensuring the purity of the EEG signal, which is crucial for the reliability and accuracy of the analysis [18, 13]. EEG signals often contain artifacts, which are undesirable interference signals that can arise from muscle movements, eye blinks, heart activity, and even interference from external electrical sources [5, 2]. These artifacts can significantly distort the true brain activity recorded by the EEG, leading to incorrect interpretations and conclusions. Therefore, effective recognition and elimination of artifacts are essential to obtain reliable results, which is especially important in the context of clinical diagnostics and precise scientific research.

*Corresponding author — anna.gajos-balinska@mail.umcs.pl

## 1.2 Temporal Convolutional Networks

Temporal Convolutional Networks (TCN) are a type of neural network specializing in processing sequential data, such as audio signals, financial data, or, as in our research, EEG signals. TCNs are a variant of convolutional networks, which have traditionally been used primarily in image processing. What distinguishes TCNs from standard convolutional networks is their specialization in analyzing time-varying data, making them particularly useful in EEG signal analysis. A key element of the TCN architecture is the use of causal convolutions, which ensure that the output at a given time point depends only on previous time points and not on future ones. This allows for the modeling of sequential phenomena while preserving the natural order of events. Moreover, TCNs utilize dilated convolutions, which enable the network to efficiently learn long-term dependencies in the data without the need to significantly increase the depth of the network. This approach allows TCNs to recognize patterns and trends in sequential data at various temporal scales [14, 12].

In the context of EEG signal analysis, TCNs offer significant advantages. Their ability to detect complex temporal and spatial patterns in EEG data allows for the effective identification of both normal brain activity and a variety of artifacts. As a result, TCNs are becoming an increasingly popular tool in brain research and clinical applications, where precision and efficiency in EEG signal analysis are of critical importance [10].

## 1.3 Independent Component Analysis

Independent Component Analysis (ICA) is an advanced statistical technique that is used in many fields – from finance to neurology. In the context of electroencephalography (EEG), ICA is used to separate the EEG signal into components that are statistically independent of each other [3]. This method allows for the isolation and identification of both physiological signal sources and various artifacts that may disrupt EEG analysis [4].

In practice, ICA analyzes the multichannel EEG signal and decomposes it into a set of components, each representing a different source of brain activity or external interference. For instance, components may represent activity related to eye movements, muscle activity, heart activity, as well as more subtle neuronal processes. Importantly, ICA allows for the separation of these different sources without the need for the physical separation of electrodes, which is particularly useful in the case of EEG signals where various sources of activity are often spatially and temporally intertwined [17, 11].

In EEG research, the application of ICA is crucial for improving the quality and interpretability of data. This allows for a more precise analysis of brain activity patterns and the detection and removal of artifacts, which is especially important in the context of diagnosing and monitoring neurological disorders. Thanks to ICA, researchers and clinicians can gain a cleaner and more reliable insight into brain activity, contributing to a better understanding of neuronal processes and more effective treatment of neurological disorders.

### 1.4 Research objective

The aim of our study is to apply Temporal Convolutional Networks (TCN) for the detection of artifacts in the EEG signal, with a particular emphasis on the use of components extracted through Independent Component Analysis (ICA). An important aspect of this is the integration of advanced signal processing methods and deep learning techniques to increase precision and efficiency in identifying artifacts in EEG data. This integration represents a significant advancement compared to traditional EEG analysis methods, which often rely on manual classification and are prone to errors.

Furthermore, TCNs, with their ability to efficiently process sequential data and recognize complex temporal patterns, offer new perspectives in EEG signal analysis. The combination with ICA, conducted on extensive data from 256 electrodes, enables a more accurate separation of artifacts from authentic brain activity, which has significant implications for the quality of diagnosis and brain research.

Additionally, this work contributes to the development of automatic EEG analysis methods, reducing the need for intensive manual labor and subjective interpretation of signals by experts. Such an approach has potential applications in various areas, from clinical diagnostics to neuroscientific research, offering faster and more reliable EEG data analysis.

## 2 Methodology

### 2.1 EEG laboratory

In our study, the use of an advanced EEG laboratory equipped with the 256-electrode system from Electrical Geodesics, Inc. (EGI) played a significant role. This highly developed EEG system offers a much greater electrode density compared to standard EEG systems, which translates into higher resolution and precision in recording brain electrical activity. Such an expanded configuration allows for more detailed tracking of neuronal processes, enabling more accurate localization of signal sources and better recognition of brain activity patterns [1].

The EGI EEG laboratory with 256 electrodes is equipped with specialized geodesic caps, which ensure optimal placement of electrodes on the scalp, key to obtaining high-quality data. Additionally, the use of the EGI system facilitates the recording of signals from various brain areas, which is particularly important in studies requiring a broad perspective on brain activity. The use of such an advanced EEG system in our study enabled a more accurate analysis and interpretation of data, which is essential for effective artifact detection and further development of EEG signal analysis methods.

The use of the 256-electrode EGI EEG system in our study on artifact detection in the EEG signal represented a significant step towards a better understanding of the complex nature of brain activity, as well as providing a solid foundation for exploring new methods and techniques in neuroscience and neurodiagnostics.

Figure 1: EEG laboratory with 256-electrode EGI System in UMCS

## 2.2 ICA for data preprocessing

In our study, we focused on analyzing EEG signals obtained in a resting state. The data were acquired from a laboratory using an EEG system equipped with 256 electrodes, utilizing a sampling rate of 250 Hz. We gathered a total of seven 20-minute EEG recordings, which formed the basis of our study. The obtained EEG data were subjected to standard filtering procedures aimed at reducing noise and external interferences.

A key element of the preliminary processing was the use of Independent Component Analysis (ICA), in accordance with the method described in the [8, 7, 6, 9]. ICA is a commonly used technique in EEG signal analysis, enabling the separation of the signal into independent components. In our case, 256 components were extracted from each of the seven EEG recordings, corresponding to the number of electrodes used.

To obtain representative samples for further analysis, we performed random sampling from the long ICA components. Up to five 1-second windows were selected from each component, coming from different time intervals. This sampling method ensured a broad representation of different signal characteristics within each component.

The next step was the classification of selected signal segments into artifacts and correct segments. This process was carried out visually, allowing for precise labeling of the data as 'clean' or containing artifacts. As a result of this classification, a total of 600 EEG signal segments were obtained, which were then used in the training and testing process of the TCN model. Figure 1 shows an example of a segment marked as correct, and Figure 2 shows an example of a segment marked as an artifact.

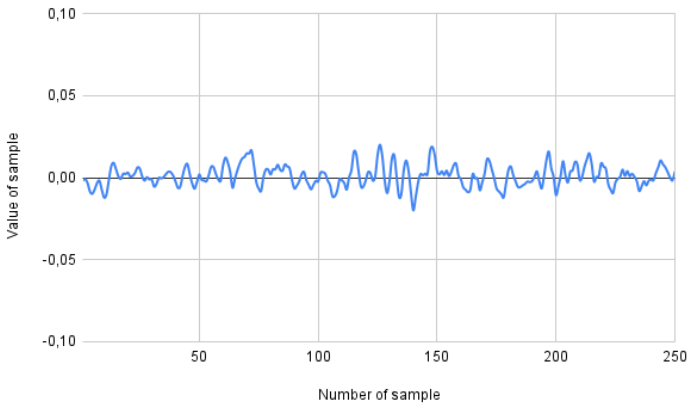



Figure 2: A segment marked as correct

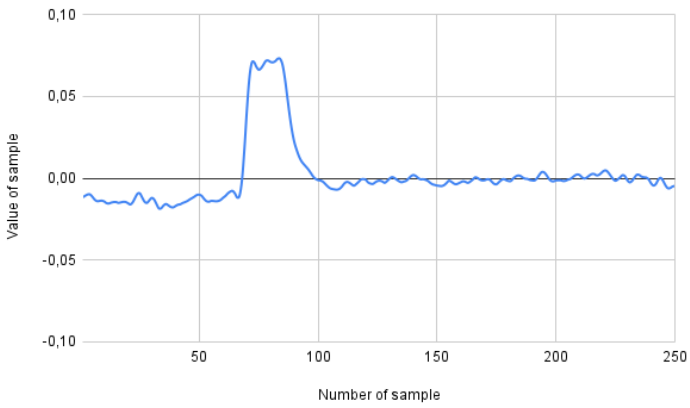



Figure 3: A segment marked as an artifact

The implementation of ICA, along with selective sampling and classification of data, allowed for effective isolation and identification of artifacts in the EEG signal. These processes provided a high-quality dataset for further analysis using Temporal Convolutional Networks, which is the subject of our further research.

### 2.3 Training data preparation

The EEG data were acquired from two different sets: 'correct' and 'artifacts', stored in CSV format. These data were divided into training and testing sets. For the purpose of the experiment, we selected 500 segments for training the Temporal Convolutional Network (TCN) model, of which 400 were correct segments and 100 were artifacts. Additionally, we prepared a test set consisting of 100 segments, including an evenly distributed 50 correct segments and 50 artifacts.

This division of training and testing data was made to ensure a balanced representation of both classes in the model learning process and during its evaluation. The use of a larger number of correct segments in the training data was intended to reflect their more frequent presence in natural EEG conditions, while the even distribution of both classes in the test data allowed for a fair assessment of the model's ability to differentiate between correct signals and artifacts.

Preliminary processing included loading the data using the Pandas library and transforming it into two-dimensional NumPy arrays, preparing them for further analysis.

### 2.4 Model description

We employed the Temporal Convolutional Network (TCN) architecture implemented in TensorFlow and Keras. The TCN architecture was configured with the following parameters: 64 filters, a kernel size of 3, and dilation rates set to [1, 2, 4, 8]. We applied 'causal' padding and used the 'relu' activation function. The TCN model was finalized with a dense layer containing one neuron.

The model was compiled using the Adam optimizer (learning rate equal to 0.001) and the MeanSquaredError loss function. During the training process, we employed callbacks such as EarlyStopping, ModelCheckpoint, and TensorBoard to monitor progress and prevent overfitting. The model was trained on the training dataset for 100 epochs with a batch size of 32.

Upon completing the training, the model was evaluated on the test dataset. The evaluation results, including accuracy and loss value, were saved. Additionally, we conducted visualization of the model's outputs on the test dataset to assess the quality of artifact classification.

## 3 Results

As part of the experiment, the TCN model was trained on 80% of the data (400 segments), with the remaining 20% (100 segments) used for testing. During the training process, techniques such as cross-validation and early stopping were employed. Cross-validation allowed for the effective utilization of the limited training data by enabling each sample to be used for both training and model validation.

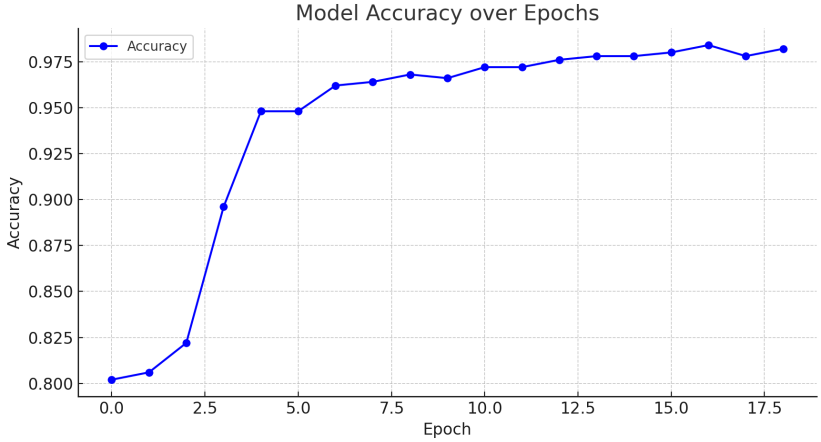



Figure 4: Model Accuracy over Epochs

Early stopping was used to monitor training progress and prevent overfitting, which is crucial for preserving the model's ability to generalize to new data.

The evaluation of the Temporal Convolutional Network (TCN) model in the context of artifact detection in EEG signals yielded promising results. The changes in accuracy and loss function values for each epoch can be observed in the Figures 5 and 4. The model achieved an accuracy of 94%, indicating its high efficiency in distinguishing between correct segments and artifacts. The loss function reached a value of 0.0657, signifying low prediction error by the model. The combination of low loss function value and high accuracy suggests that the TCN model was able to effectively learn to differentiate artifacts from normal EEG signal segments, even with a relatively limited amount of training data.

Additionally, the analysis of individual predictions made by the model for 100 test samples (50 correct and 50 containing artifacts) demonstrates that the model effectively identifies the majority of artifacts, with prediction values significantly differing from those for correct segments. In the case of correct segments, prediction values are mostly close to zero or have low absolute values, indicating correct classification. However, for segments with artifacts, these values are considerably higher, often exceeding 1, confirming their accurate identification as artifacts.

Such results highlight the potential of TCN in the context of EEG signal analysis, particularly in applications where fast and accurate artifact detection is crucial. However, it is essential to note isolated cases where the model may have misclassified some segments. This observation underscores the need for further optimization and adaptation of the model, especially concerning increased diversity in the training data.

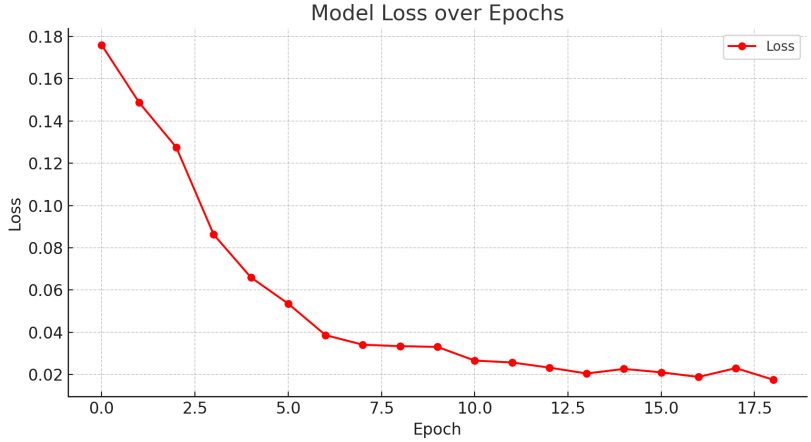



Figure 5: Model Loss over Epochs

# 4 Discussion

Analyzing the results, it is evident that TCN demonstrates exceptional effectiveness in detecting artifacts that traditional methods may overlook. Particularly in cases of subtle artifacts, where differences from normal EEG signals are minimal, TCN exhibits the ability for precise classification. The application of deep convolutional networks with dilation mechanisms allows for the effective recognition of patterns in sequential data, which is crucial in EEG signal analysis.

It is worth noting that the TCN model also surpasses traditional methods in terms of automation and processing speed, which is of significant importance in practical applications, especially in clinical environments where rapid and accurate artifact detection can be critical.

One of the main limitations of our study is the relatively small number of data segments used to train the TCN model. Using 500 segments, including 400 correct and 100 artifacts, may not fully capture the complexity and diversity of clinical scenarios in which EEG signals are collected. The limited amount of training data restricts the model's ability to learn subtler patterns, potentially affecting its capacity to generalize to new, unknown data.

In future research, it will be essential to increase the available training data. Using a larger dataset would enable a better utilization of TCN network potential, particularly in the context of recognizing more complex and subtle artifacts in EEG signals.

There is also potential for experimentation with various methods for selecting segments for training and testing sets. Testing the model on different combinations of correct and artifact segments can provide valuable insights into its performance in various scenarios.

Additionally, modifying and optimizing TCN network parameters is possible. Experimenting with the number of filters, kernel size, dilation rates, activation functions, as well as other architectural elements, could significantly impact the

model's ability to learn and classify signals. Such adjustments can lead to increased model accuracy, especially in the context of detecting subtle artifacts that are challenging to identify.

Finally, in future research, it is worth considering the application of alternative deep learning methods and comparing their effectiveness with TCN. Such an analysis would provide a better understanding of the advantages and limitations of different approaches in the context of EEG signal analysis.

# 5 Conclusions and future works

The study employing the Temporal Convolutional Network (TCN) for artifact detection in EEG signals has yielded promising results. The TCN model demonstrated a high accuracy of 94% despite a relatively limited training dataset, emphasizing its effectiveness in distinguishing between correct segments and artifacts. The low loss function value indicates the model's precision in signal classification, which is crucial in practical applications within EEG analysis. These results affirm TCN's potential as a tool for enhancing diagnostic processes and brain research, where precise artifact detection is essential.

Considering the current findings and limitations, future work should focus on expanding the training dataset. A larger and more diverse dataset will facilitate better model generalization and enable a more thorough analysis of subtle artifacts. Furthermore, experimenting with various TCN network parameter configurations, including the number of filters, kernel size, and dilation rates, may further enhance model performance. It is also worthwhile to consider comparing TCN with other advanced deep learning methods to identify the best approaches for EEG signal analysis. These steps have the potential to lead to significant advancements in the field of neurological signal processing and open new avenues for brain function research.

# Acknowledgements

This work was supported by the EU Research and Innovation programme Horizon 2020 under grant agreement No. 952357 – REINITIALISE (Preserving fundamental rights in the use of digital technologies for e-health services). The authors would like to thank the European Commission for enabling the funding. Additionally, special thanks goes to supervisor and co-author of the publication prof. Bart Vanrumste from KU Leuven.

# References

[1] Netstation acquisition technical manual. documentation, egi, 2011.

[2] Anton Albajes-Eizagirre, Laura Dubreuil Vall, Ibanez-Soria David, Alejandro Riera, Aureli Soria-Frisch, Stephen Dunne, and Giulio Ruffini. *EEG/ERP analysis: methods and applications.* 10 2014.

[3] Glen D Brown, Satoshi Yamada, and Terrence J Sejnowski. Independent component analysis at the neural cocktail party. *Trends in neurosciences*, 24(1):54–63, 2001.

[4] Arnaud Delorme, Terrence Sejnowski, and Scott Makeig. Enhanced detection of artifacts in eeg data using higher-order statistics and independent component analysis. *Neuroimage*, 34(4):1443–1449, 2007.

[5] Cheryl L Dickter and Paul D Kieffaber. *EEG methods for the psychological sciences*. Los Angeles, 2014.

[6] Anna Gajos-Balińska, Grzegorz M. Wójcik, and Przemyslaw Stpiczyński. Concept of independent component analysis algorithm parallelisation. In *Proceedings of Cracow Grid Workshop*, CGW'15, pages 55–56, 2015.

[7] Anna Gajos-Balińska, Grzegorz M. Wójcik, and Przemyslaw Stpiczyński. Parallel independent component analysis algorithm - performance comparison for eeg signal. In *Proceedings of Cracow Grid Workshop*, CGW'17, 2017.

[8] Anna Gajos-Balińska, Grzegorz M. Wójcik, and Przemyslaw Stpiczyński. High performance optimization of independent component analysis algorithm for eeg data. *Lecture Notes in Computer Science*, 10777:495–504, 2018.

[9] Anna Gajos-Balińska, Grzegorz M Wójcik, and Przemysław Stpiczyński. Cooperation of cuda and intel multi-core architecture in the independent component analysis algorithm for eeg data. *Bio-Algorithms and Med-Systems*, 16(3), 2020.

[10] Zhipeng He, Yongshi Zhong, and Jiahui Pan. An adversarial discriminative temporal convolutional network for eeg-based cross-domain emotion recognition. *Computers in Biology and Medicine*, 141:105048, 2022.

[11] Aapo Hyvärinen and Erkki Oja. Independent component analysis: algorithms and applications. *Neural networks*, 13(4):411–430, 2000.

[12] Daniel Jarrett, Jinsung Yoon, and Mihaela van der Schaar. Dynamic prediction in clinical survival analysis using temporal convolutional networks. *IEEE Journal of Biomedical and Health Informatics*, 24(2):424–436, 2020.

[13] Aleksandra Kawala-Janik, Waldemar Bauer, Amir Al-Bakri, Chase Haddix, Rajamanickam Yuvaraj, Katarzyna Cichon, and Wojciech Podraza. Implementation of low-pass fractional filtering for the purpose of analysis of electroencephalographic signals. In *Conference on Non-integer Order Calculus and Its Applications*, pages 63–73. Springer, 2017.

[14] Colin Lea, Michael D. Flynn, Rene Vidal, Austin Reiter, and Gregory D. Hager. Temporal convolutional networks for action segmentation and detection, 2016.

[15] Emilia Mikołajewska and Dariusz Mikołajewski. Integrated it environment for people with disabilities: a new concept. *Open Medicine*, 9(1):177–182, 2014.

[16] Emilia Mikołajewska and Dariusz Mikołajewski. The prospects of brain–computer interface applications in children. *Open Medicine*, 9(1):74–79, 2014.

[17] M Ungureanu, C Bigan, R Strungaru, and V Lazarescu. Independent component analysis applied in biomedical signal processing. *Measurement Science Review*, 4(2):18, 2004.

[18] Grzegorz Wojcik. *Selected methods of quantitative analysis in electroencephalography*, pages 35–54. 04 2020.

# Logistic Regression Model for the Credibility Evaluation Dense-Array EEG Signal Classification

Piotr Schneider*
Bart Vanrumste
Grzegorz M. Wojcik

## 1 Introduction

Credibility evaluation in society involves the assessment of the trustworthiness, reliability, and authenticity of various entities, including individuals, institutions, information sources, and public figures [6, 3, 7]. In a broader societal context, credibility evaluation plays a crucial role in shaping public opinion, decision-making processes, and the overall functioning of communities.

Society relies on various sources of information, such as news outlets, websites, and social media. Credibility evaluation involves assessing the reliability and objectivity of these sources. Factors like journalistic standards, fact-checking practices, and editorial transparency contribute to the credibility of media organizations.

Individuals or institutions are often evaluated based on their expertise and authority in a particular field. Experts and authoritative figures are generally considered more credible, especially when their qualifications and experience align with the subject matter [2, 4].

Credibility is enhanced when individuals, organizations, or institutions are transparent about their actions, decision-making processes, and financial dealings. Openness and accountability contribute to trust in society.

Past behavior and a track record of reliability contribute to credibility. Individuals or organizations with a positive reputation for honesty, ethical conduct, and successful outcomes are often considered more credible.

Credibility evaluation involves assessing the trustworthiness, reliability, and authenticity of various entities, such as individuals, institutions, and information sources. There are several methods and approaches to evaluate credibility across different contexts [7].

*Corresponding author — piotr.schneider@mail.umcs.pl

In the scientific community, peer review is a standard method for evaluating the credibility of research studies. Other experts in the field review and assess the methodology, results, and conclusions of a study before it is published in a scientific journal.

Assessing an individual's credentials, qualifications, and expertise in a particular field is a common method for evaluating credibility. Advanced degrees, professional certifications, and relevant experience contribute to perceived expertise.

But how to evaluate credibility in more sophisticated, quantitative and scientific way instead of trusting documents or other?

Gaining a more profound insight into the cognitive processes underlying the assessment of both source and message credibility will move us closer to the ambitious objective of developing a diagnostic approach to evaluate the credibility of online content using EEG. This method, devoid of biases, has the potential to unveil the comprehensive effects of disinformation on the human brain.

The aim of this paper is to investigate the effectiveness of machine learning model, in this case Logistic Regression, in classification of the electroencephalographic resting state signal collected from subjects doing some credibility evaluation task.

# 2 Material and methods

## 2.1 The Lab

Our practical investigations utilized leading EEG devices, and the laboratory setup was a comprehensive and compatible system provided by Electrical Geodesic Systems (EGI). We employed a dense array amplifier to record cortical activity at a frequency of up to 500 Hz across 256 channels using HydroCel GSN 130 Geodesic Sensor Nets from EGI. Additionally, the EEG Laboratory incorporated the Geodesic Photogrammetry System (GPS), which creates a model of the subject's brain based on calculated size, proportion, and shape. This is achieved using 11 cameras positioned at the corners, accurately overlaying all computed activity results onto this model. The amplifier operated with Net Station 4.5.4 software, GPS was controlled by Net Local 1.00.00, and GeoSource 2.0. Gaze calibration, elimination of eye blinks, and saccades were accomplished through an eye-tracking system facilitated by SmartEye 5.9.7. Event-related potential (ERP) experiments were designed within the OpenSesame 3.2.8 environment.

The removal of artifacts, including eye blinks and saccadic eye movements, was carried out using scripts integrated into the EGI system software. While this software is not open source, detailed information about its functionality is broadly available in the Waveform Tools Technical Manual [1].

## 2.2 The Cohort

Our study involved 73 male subjects who were right-handed and aged between 21 and 22 years (average 21.3, standard deviation 0.458). Among these, 40 participants had EEG signals recorded adequately for both stages of the experiment. The signal-to-noise ratio (SNR) was deemed satisfactory, all tasks (screens) were

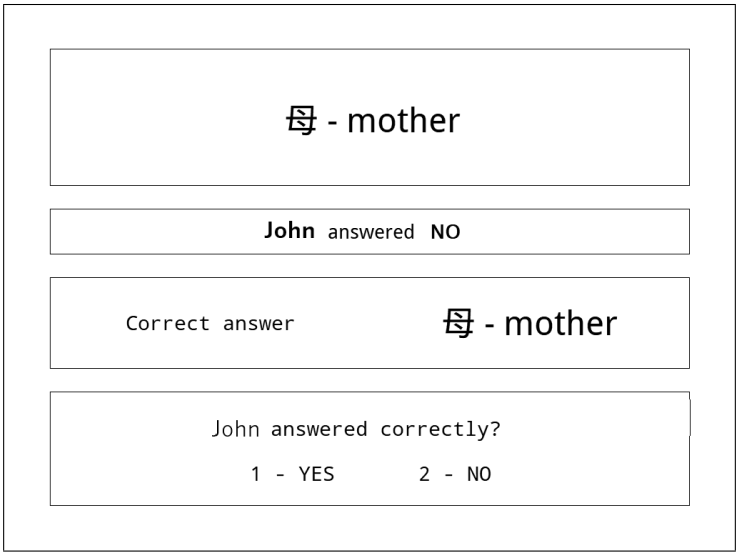



Figure 1: Typical screen shown to participant in stage 1 of the experiment

successfully completed, and a sufficient number of epochs were recorded, allowing for subsequent source localization.

To conduct the experiment, we obtained approval from the University's Bioethical Commission (MCSU Bioethical Commission permission granted on 13.06.2019).

## 2.3 The Experiment

Participants who lacked knowledge of Japanese (confirmed through a pre-recruitment questionnaire) were informed that they would be reviewing the outcomes of an authentic Japanese language test taken by other students. Specifically, participants were instructed that they would be examining the responses of three students: S1 (Bruno), S2 (Cesar), and S3 (John).

The primary objective of the experiment was to replicate a comprehensive credibility evaluation task while maintaining controlled knowledge conditions. Participants were presented with questions and answers from either S1, S2, or S3, alongside the correct answer. Although participants possessed complete knowledge of the presented information, their initial familiarity with the subject matter (Japanese language) was nonexistent [5].

A representative screen from the first stage of the experiment is illustrated in **??**, featuring four sections. All participants received prior instruction on the screen layout before commencing the experiment.

The initial two sections of the screen are derived from the test, where students were tasked with evaluating the proposed meanings of certain Kanji signs as either true or false.

In the top section of the screen, a Kanji sign is presented along with the corresponding question about its meaning in the native language of the participants.

The second section of the screen contains the responses of either student S1, S2, or S3. These responses indicate whether, according to the tested student, the meaning of the Kanji sign in the first section is true or false.

The subsequent two sections include a hint and a task for the participants.

In the third section, the correct, dictionary meaning of the Kanji sign from Section 1 is provided.

The fourth section presents the task for the participant: to assess whether the response of the tested student is correct or incorrect.

Throughout the first stage of the experiment, participants encountered a total of 408 screens: 136 for each student (S1, S2, and S3). The first 136 screens featured responses from S1, the next 136 screens featured responses from S2, and the final 136 screens featured responses from S3 on the simulated test. Prior to the assessment series for each student (S1, S2, and S3), participants were shown a screen informing them that they would now be assessing the student in question.

The responses presented to participants from students S1, S2, and S3 were deliberately selected with specific characteristics:

- Student S1 was intentionally portrayed as a weak student, having only 25% of correct responses during the test.
- Student S2 was depicted as an average student, with 50% of his responses being correct.
- Student S3 was portrayed as a proficient student, as 75% of his responses were correct.

It's important to note that during the experiment, participants possessed perfect knowledge of the correct answers but lacked any prior information about the performance of the individual students. Consequently, participants made their credibility evaluations based solely on message credibility. This experimental design ensured a scenario where, after assessing 136 responses from each of the students (S1, S2, and S3), participants gained awareness of each student's proficiency level—identifying who performed well, who performed poorly, and who was of average performance. We confirmed this condition by querying participants about their opinions on the proficiency levels of students S1, S2, and S3, using a specially presented screen after the conclusion of the experiment. The experiment is explained in more detail in [5].

# 3 Results

## 3.1 Effectiveness of performance in the initial phase of the experiment

In the initial phase of the study, participants were tasked with acquiring an understanding of the credibility assessments associated with three imaginary students: S1, S2, and S3. Following the completion of this first phase, participants were requested to evaluate the performance of students S1, S2, and S3. The objective was to determine if participants had gained comprehensive knowledge about the historical performance of these students. Only those participants who demonstrated such

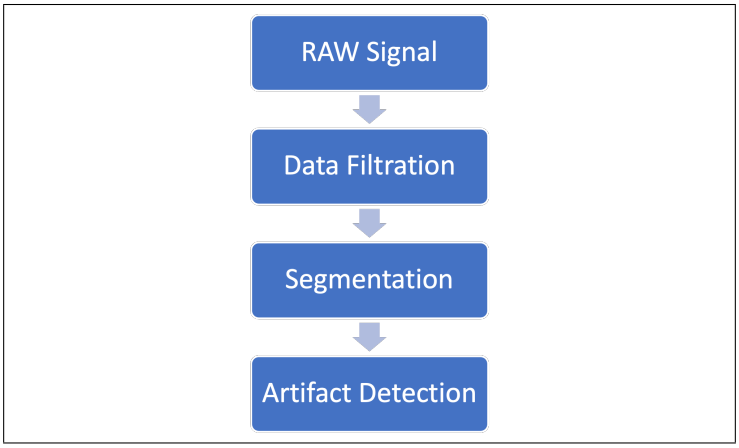



Figure 2: Preprocessing steps

knowledge were permitted to proceed to the second stage, denoted as P2. Among the subset of 36 participants with a sufficiently high-quality signal, 14 successfully and appropriately assessed the knowledge levels of the fictitious students S1, S2, and S3. The data was analyzed using **Python 3.10.13** and **scikit-learn 1.3.1** libraries.

## 3.2 Preparing data for machine learning models

The data obtained from the EEG experiment were recorded and analyzed using the MNE tool and the Python language. The pipeline process for the preprocessing step is depicted in Fig. 2.

Following the preprocessing, we acquired segmented data. For each segment, spectrograms were calculated, with a data frequency of 250 Hz and segment lengths of 1 second. Each segment has dimensions of 125 frequencies by 256 electrodes.

## 3.3 Machine learning models

In the previous article [5], we achieved commendable results using Brodmann's area, with an accuracy (ACC) exceeding 0.7. In the current study, we employed data derived from spectrograms and aimed to construct a model to assess the potential for improved results. To this end, we selected the Logistic Regression algorithm. For logistic regression, we implemented the Recursive Features Elimination (RFE) algorithm to identify the optimal combination of features that yield favorable results. Our objective is to achieve an accuracy surpassing 0.7.

During the initial phase of the experiment, participants' credibility assessments were categorized into two primary cases: MC and MNC.

The objective of constructing a machine learning model is to attempt to predict whether the participant's credibility evaluation falls into the categories of MC or MNC. In this context, MC was considered the positive class. The model outcomes are detailed in Table 1.

Table 1: Results acquired during the data preparation phase for the model aimed at classifying instances of MC and MNC

| Name | Value |
|---|---|
| Time interval | 1 second segment |
| Total avg. segemnts (each class) | 30 |
| Independent variables (RFE algorithm) | 86 electrodes |
| Independent variables (model) | 86 electrodes |
| mean ACC | 0.77 |
| Standard deviation | 0.0106 |
| Training data | 3024 observations |
| Test data | 504 observations |

Table 2: Metrics assessing the logistic regression model's performance in forecasting the credibility of messages during the initial stage of the experiment

| | |
|---|---|
| Accuracy | 0.67 |
| Precision | 0.66 |
| Recall | 0.70 |
| F1 | 0.68 |

The 86 electrodes was obtained from averaged data. The averaged data were obtained by separately averaging segments for each class (MC and MNC) within each subject. The model was build using only 86 from 256 electrodes. We employed a logistic regression estimator with 7-fold cross-validation. The estimator utilized default values, with the solver set to 'newton-cholesky'. This configuration yielded an accuracy (ACC) of 7. From these individual results, we computed the mean ACC and standard deviation (see Table 2).

The quality measures of the classifier are presented in Table 2.

The presented results substantiate the feasibility of constructing a robust model for predicting message credibility, relying on the outcomes of the first part of our experiment.

# 4 Conclusions

Credibility evaluation in society is a dynamic process that evolves based on changing circumstances, societal norms, and the information landscape. In an era of digital communication and social media, the rapid dissemination of information makes it crucial for individuals and organizations to actively manage and maintain their credibility to foster trust within the broader community.

In the preceding article [5], the focus was on utilizing Brodmann's areas as input for the classification algorithm. In this article, our attempt is to construct a model by employing spectrograms calculated from the raw signals obtained from

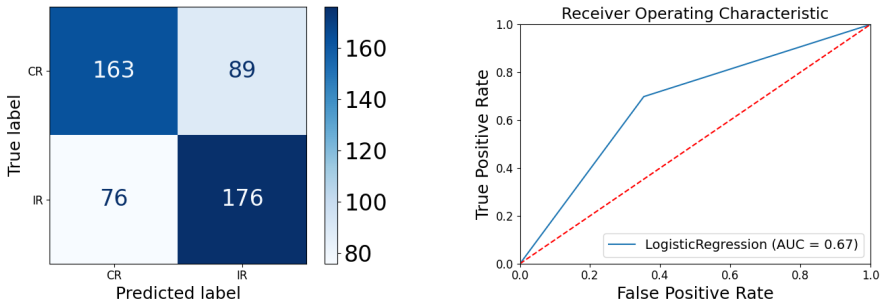



Figure 3: Confusion matrix (left side) and the ROC curve (right side) for the model classifying message credibility evaluations, specifically distinguishing between the cases of MC and MNC)

electrodes.

Our findings validate that employing source localization algorithms (such as sLORETA) and machine learning classifiers enables the prediction of message credibility evaluation, both with and without perfect knowledge. By constructing a model involving a finite number of variables, we achieved an accuracy level of approximately 0.77.

In the subsequent phase, we plan to expand our participant pool to explore the potential of constructing universal models capable of predicting decisions made by individuals outside the current cohort. Preliminary findings from our initial efforts have shown promise. Evaluating our approach with data collected from new subjects, entirely distinct from the training and validation sub-cohorts, yielded an average accuracy of 0.67 for first stage of the experiment. These results will be detailed in upcoming reports.

For future investigations, incorporating convolutional neural networks and leveraging deep learning methods is anticipated. These advanced techniques have the potential to enhance the efficiency of standard AI classifiers by several percentage points.

## Acknowledgements

This work was supported by the EU Research and Innovation programme Horizon 2020 under grant agreement No. 952357 – REINITIALISE (Preserving fundamental rights in the use of digital technologies for e-health services). The authors would like to thank the European Commission for enabling the funding. Additionally, special thanks goes to co-author of the publication Prof. Bart Vanrumste from Katholieke Universiteit, Leuven without whom development of this research would be impossible (PS, GMW).

# References

[1] EGI. *Net Station Waveforms Tools Technical manual.* Electrical Geodesics, Inc., 2006.

[2] Michal Kakol, Michal Jankowski-Lorek, Katarzyna Abramczuk, Adam Wierzbicki, and Michele Catasta. On the subjectivity and bias of web content credibility evaluations. In *Proceedings of the 22nd international conference on world wide web*, pages 1131–1136. ACM, 2013.

[3] Michal Kakol, Radoslaw Nielek, and Adam Wierzbicki. Understanding and predicting web content credibility using the content credibility corpus. *Information Processing & Management*, 53(5):1043–1061, 2017.

[4] Maria Rafalak, Katarzyna Abramczuk, and Adam Wierzbicki. Incredible: Is (almost) all web content trustworthy? analysis of psychological factors related to website credibility evaluation. In *Proceedings of the 23rd international conference on world wide web*, pages 1117–1122. ACM, 2014.

[5] Piotr Schneider, Grzegorz M Wójcik, Andrzej Kawiak, Lukasz Kwasniewicz, and Adam Wierzbicki. Modeling and comparing brain processes in message and earned source credibility evaluation. *Frontiers in Human Neuroscience*, 16:808382, 2022.

[6] Shawn Tseng and BJ Fogg. Credibility and computing technology. *Communications of the ACM*, 42(5):39–44, 1999.

[7] Adam Wierzbicki. *Web Content Credibility.* Springer, 2018.

# Application of Convolutional Neural Networks to Classification of Introversion and Extraverison Biomarkers in Dense-Array EEG Signal

Filip Postępski*
Bart Vanrumste
Grzegorz M. Wojcik

## 1 Introduction

Introversion and extraversion represent two key dimensions of personality, characterizing individuals based on their social tendencies and energy orientation. Introverts are commonly described as individuals who draw their energy from internal thoughts and reflections, thriving in solitary environments. These individuals often exhibit reserved and contemplative behaviors, preferring smaller social gatherings or individual pursuits. Introverts may find excessive social interaction draining and tend to recharge through solitude and introspection. They are often thoughtful and observant, thriving in settings that allow for deep focus and meaningful one-on-one connections.

On the flip side, extraverts are characterized by their outgoing and sociable nature. These individuals gain energy from external stimuli and thrive in social situations. Extraverts are typically seen as talkative, energetic, and assertive, seeking out a variety of social engagements to satisfy their need for interaction and stimulation. They often enjoy large gatherings, parties, and group activities, where they can express themselves and connect with others. Extraverts tend to be more spontaneous, adaptable, and outwardly expressive, embracing a more visible presence in social dynamics. The contrast between introversion and extraversion represents a spectrum of personality traits, with most individuals falling somewhere along this continuum.

The Myers–Briggs Type Indicator (MBTi) is a widely used personality assessment tool that incorporates introversion and extraversion as fundamental dimen-

*Corresponding author — filip.postepski@mail.umcs.pl

Table 1: Classification accuracy for each fold

| Fold no. | Validation ACC | Validation loss |
|---|---|---|
| 1 | 0.7527 | 1.1752 |
| 2 | 0.6499 | 2.1663 |
| 3 | 0.8250 | 0.5754 |
| 4 | 0.6388 | 1.9488 |
| 5 | 0.6917 | 3.088 |
| AVG | 0.7116 | 1.7907 |
| Std. | 0.0775 | 0.9624 |

sions. In the MBTi, introversion and extraversion reflect an individual's preferred orientation toward the external world. Introverts, designated by the "I" in the MBTi, are inclined to focus on their inner world of thoughts and ideas, finding energy and rejuvenation in solitary activities. They tend to be reflective, reserved, and contemplative. On the other hand, extraverts, denoted by the "E," direct their energy outward, finding stimulation and vitality through engagement with the external environment. Extraverts are often sociable, expressive, and energized by social interactions. The MBTi recognizes that individuals exhibit varying degrees of introversion and extraversion, providing a nuanced understanding of how people prefer to interact with and derive energy from the world around them [5].

Mental workload measurement is a crucial aspect of assessing cognitive demands and resource allocation during various tasks, particularly in the context of human-machine interactions and complex work environments. Researchers and practitioners employ diverse methodologies to quantify mental workload, often combining subjective and objective measures. Subjective assessments involve self-reported ratings or questionnaires, where individuals express their perceived workload levels. This can provide valuable insights into the cognitive demands experienced by individuals during a task. Objective measures, on the other hand, may include physiological indicators such as heart rate variability, eye tracking, and neuroimaging techniques like electroencephalography (EEG). These measures offer an empirical and physiological perspective on mental workload, complementing subjective assessments and providing a more comprehensive understanding of cognitive resource utilization [7, 2].

Efforts to improve mental workload measurement also involve the development of advanced technologies and algorithms. For instance, machine learning approaches can analyze behavioral patterns, physiological responses, and task performance to infer mental workload levels objectively. The integration of real-time monitoring and adaptive systems aims to enhance workload management by dynamically adjusting task demands to optimize individual performance. As the understanding of mental workload continues to evolve, the integration of both subjective and objective measures offers a multifaceted approach for a comprehensive evaluation of cognitive demands, contributing to the design of more efficient and user-friendly systems and work environments [9].

The aim of this paper is to find out if it is possible to classify extraversion and introversion using only raw EEG signal recordings of the mental workload tasks.

From different machine learning methods CNN were selected for this research as it was proven that models based on them can accurately classify different types of data like images or signals, including EEG data. Personality traits were classifed with success using 2D convolutional network EEGNet. However it is also posible to use 1D convolutional neural networks, as EEG data can be treated as time-series data [3, 6, 1].

# 2 Materials and Methods

## 2.1 The Lab

All EEG recordings were acquired utilizing a 256-channel dense-array EEG amplifier equipped with a HydroCel GSN (geodesic sensor net) 130, manufactured by Electrical Geodesic Systems (EGI) located at 500 East 4th Ave. Suite 100, Eugene, OR 97401, USA. The sampling frequency for the recordings was set at 250 Hz. The amplifier functioned in conjunction with Net Station 4.5.4 software and SmartEye 5.9.7 for tasks such as gaze calibration and the removal of artifacts caused by eye-blinking or saccadic movements. Additionally, the laboratory featured a geodesic photogrammetry system (GPS), operated using Net Local 1.00.00 and GeoSource 2.0.

## 2.2 The Cohort

For this study cohort of 20 subjects was selected from the group of 27 recruited. All participants were right-handed, short-haired, males of age 19 - 24 years old. They were all students of Computer Science studies. Each subject signed written agreement before taking part in the experiment. They were informed about purpose of the EEG experiment and the way it will be conducted. In the first step they all filled MBTi questionnaire. 10 people were classified as introverts and 10 were classified as extroverts. In the next step all participants underwent EEG acquisition procedure.

### 2.2.1 Inclusion criteria

Participants in the study were required to fall within the age range of 17 to 24, aligning with the typical age demographic of computer science students at the university where the experiment was conducted. Specific criteria included being short-haired, right-handed men, as longer hair could interfere with signal recording, introducing unwanted noise. Given the relatively low number of women in computer science at the time, creating a balanced cohort that included an equal distribution of left-handed and right-handed individuals, as well as both genders, would have been challenging. Additionally, the majority of female computer science students had long hair. Notably, previous studies highlighted differences in electroencephalograms between men and women, and efforts were made to ensure a relatively equal response within the cohort.

The selection process also considered the potential impact of lateralization, assuming that handedness could play a significant role in classification. All chosen

participants belonged to the demographic of white men of Polish nationality or citizenship who were fluent in Polish.

Health-related inclusion criteria encompassed being in good health, not using prescribed medication, soft drugs, or hard drugs, having no medical treatment history in the year following the study, and not experiencing chronic diseases, including chronic fatigue syndrome, cancer, or any other physical or mental disorders. Participants were required to have the ability to attend study appointments without specific technological requirements.

Furthermore, participants were nonsmokers and were instructed to abstain from consuming alcohol or any medications for at least 72 hours before participating in the experiment.

### 2.2.2 Exclusion criteria

Individuals falling below the age of 17 or exceeding 24 years, those who were left-handed, possessed long hair, or were women were automatically excluded from the cohort recruitment process for the reasons elucidated earlier.

Moreover, participants who lacked fluency in the Polish language were excluded, considering that the gastrointestinal (GI) session and mental tasks were conducted and formulated in Polish, respectively. To ensure study replication, it is recommended to select the same language for GI sessions, mental tasks, and cohort members.

Candidates exhibiting even minor illnesses (such as the flu, cold, or a running nose) were excluded from the cohort recruitment process. Those taking prescribed medications, soft drugs, or hard drugs were also ineligible for recruitment.

Further exclusion criteria involved candidates with a medical treatment history within one year following the study, as well as those with chronic diseases, encompassing chronic fatigue syndrome, cancer, or any other diagnosed physical or mental disorders.

Candidates unable to attend study appointments were likewise ineligible for inclusion in the cohort.

## 2.3 EEG signal acquisition and preprocessing

All the EEG recordings were done in EEG Laboratory o Maria Curie-Skłodowska University in Lublin. For this 256-channel EGI EEG cap was used. All the recordings were collected with sampling rate of 250Hz. Software used for signal recordings was EGI Net Station. Each recording was 20 minutes long and consisted of 5 mental tasks requiring recalling of information which included: Zodiac signs, all the neighbour countries of Poland, capital cities of European countries, names of polish regions (voivodsips), all of the USA states. Instruction for each task was spoken on the recording and then there was 5 minutes long period of silence for information recalling. Participants were told that they will have to write down all of the recalled information after the EEG experiment. All of the signals were recorded in lying position with participant's eyes closed and light turned off. That way influence of the muscle artifacts or line noise was reduced. For signal preprocessing mne 1.3.0 Python toolkit was used. Only two steps of signal preprocessing were done. Bad channels were interpolated using RANSAC algorithm implemented

in autoreject toolkit. Then signal was filtered using band-pass filter from 1 to 45 Hz. No manual feature extraction was done.

### 2.4 Data set preparation

Prepared signal was cropped beginning from 11-14 min. Which resulted in signals of 180 s for each subject. Time period selection was arbitrary. But during this period there was silence in the recording. Next, cropped signal was split into 2-s segments which gave 90 samples of EEG signal per subject. Each sample had 21 channels selected from 256 electrodes based on 10-20 international system [4]. As the result final data set for machine learning consisted of 1800 samples. Validation data set consisted of 360 samples and training dataset contained 1440 samples. Each sample was labeled Extroversion or Introversion according to MBTi questionnaire filled by each subject.

### 2.5 Convolutional Neural Networks

Convolutional Neural Networks (CNNs) are a class of deep neural networks primarily designed for analyzing visual data like images and videos. They use a specialized architecture that incorporates convolutional layers to automatically and adaptively learn features from the input data. CNNs employ filters that slide across the input data, enabling the network to identify spatial hierarchies of patterns, gradually extracting complex features [3].

Final architecture consisted of 6 1D convolution layers followed by LeakyReLU activation layer. Max Pooling layer was used to reduce size of processed data after each feature extraction after each activation layer except the last one before Flatten layer. Purpose of the Flatten layer is to turn all feature maps into one tensor for two dense layers which are responsible for binary classification. Between those two layers there is a dropout layer used as regularization method [8, 3].

The CNN model was formulated using the Keras and TensorFlow 2.12 libraries in Python 3.7. Testing was conducted on an Intel i7-based system featuring 32GB of RAM, accompanied by an Nvidia GeForce GTX 970 with 4GB of RAM. Ubuntu 18.04 LTS served as the operating system, and it's important to note that no components in the setup were overclocked.

## 3 Results

Using 5-fold cross-validation designed architecture achieved classification accuracy of 71.16%. Results for each fold are shown in Table 1 with loss. For the best fold 82.50% of accuracy was obtained while for the worst - 63.88%. While standard deviation of the accuracy can be considered low, standard deviation for loss function is relatively high. It can be the result of differences between subsequent learning samples.

It can be seen in Figure 1 that cross-validated accuracy is not stable at the beginning of the learning process. It achieves 70% in the 41st epoch and becomes stable with decreasing standard deviations in following epochs.

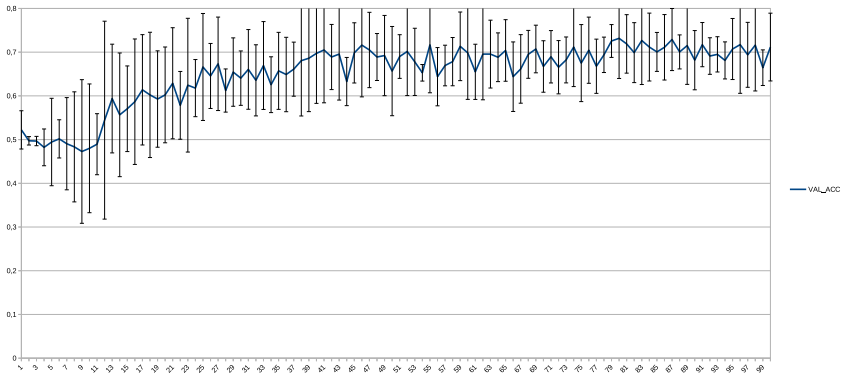



Figure 1: Validation accuracy averaged over 5 folds with standard deviation

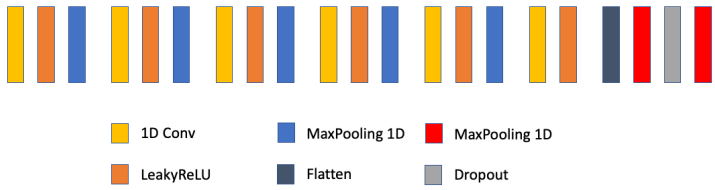



Figure 2: Proposed 1D-CNN architecture

Results achieved by this architecture were possible while using specific parameters in each layer. Filter sizes in next convolutional layer were increasing by 2 - starting from 16 in the first layer, finishing with 256 in the last one. Kernel size of 3 was determined to work best in this model. Best pooling size in Max Pooling layer was determined to be 2. Discussed architecture of the network is presented in Figure 2 .

# 4 Conclusions

Distinguishing between introversion and extraversion is crucial in understanding and appreciating the diversity of human personality and behavior. These personality traits, identified by renowned psychologists like Carl Jung and incorporated into various personality assessment models, including the Myers–Briggs Type Indicator (MBTi), play a significant role in shaping how individuals interact with the world around them.

The future of Convolutional Neural Networks (CNNs) in biomedical signal processing holds significant promise and potential for transformative advancements in healthcare and medical research. CNNs, a type of deep learning architecture, have demonstrated remarkable capabilities in processing and analyzing complex data, making them well-suited for handling biomedical signals.

The role of mental workload in the decision-making process is of paramount importance as it significantly influences the quality and efficiency of decisions made by individuals. Mental workload refers to the cognitive resources and processing demands required to perform a task or make decisions.

It was shown that Convolutional Neural Networks can be used with for raw EEG data-based classification of extroversion and introversion. Recordings of mental workload tasks were used for this purpose. The described architecture achieved 71.16% of 5-fold cross-validated accuracy. It is a good starting point for further research. It still needs to be investigated how the specific data and network architecture will affect classification accuracy.

A more nuanced comprehension of the brain mechanisms governing the decision-making process, tailored to individual personality types, holds the promise of future advancements in the development of user-friendly brain-computer interfaces. This understanding paves the way for the creation of interfaces that are optimized to accommodate the distinct preferences of introverted or extraverted individuals, enhancing the overall efficiency and usability of these interfaces in supporting the decision-making journey.

## Acknowledgements

This work was supported by the EU Research and Innovation programme Horizon 2020 under grant agreement No. 952357 – REINITIALISE (Preserving fundamental rights in the use of digital technologies for e-health services). The authors would like to thank the European Commission for enabling the funding. Additionally, special thanks goes to co-author of the publication Prof. Bart Vanrumste from Katholieke Universiteit, Leuven without whom development of this research would be impossible (FP, GMW).

## References

[1] Veronika Guleva, Alessandra Calcagno, Pierluigi Reali, and Anna Maria Bianchi. Personality traits classification from eeg signals using eegnet. In *2022 IEEE 21st Mediterranean Electrotechnical Conference (MELECON)*, pages 590–594, 2022.

[2] Allen L Hammer. *MBTI applications: A decade of research on the Myers-Briggs Type Indicator*. Consulting Psychologists Press, 1996.

[3] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. *Commun. ACM*, 60(6):84–90, may 2017.

[4] Phan Luu and Thomas Ferree. Determination of the hydrocel geodesic sensor nets' average electrode positions and their 10–10 international equivalents. *Inc, Technical Note*, 1(11):7, 2005.

[5] Isabel Briggs Myers. *A Guide to the Development and Use of the Myers-Briggs Type Indicator: Manual*. Consulting Psychologists Press, 1985.

[6] Shu Lih Oh, Jahmunah Vicnesh, E. J. Ciaccio, Rajamanickam Yuvaraj, and U Rajendra Acharya. Deep convolutional neural network model for automated diagnosis of schizophrenia using eeg signals. *Applied Sciences*, 9(14), 2019.

[7] Hongquan Qu, Yiping Shan, Yuzhe Liu, Liping Pang, Zhanli Fan, Jie Zhang, and Xiaoru Wanyan. Mental workload classification method based on eeg independent component features. *Applied Sciences*, 10(9):3036, 2020.

[8] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.*, 15(1):1929–1958, jan 2014.

[9] Anukriti Yadav, Deepak Kumar, and Yasha Hasija. Behaviour analysis using machine learning algorithms in health care sector. In *2023 International Conference on Advancement in Computation & Computer Technologies (InCACCT)*, pages 877–880, 2023.

# Comparison of Different Approaches to SARS-CoV-2 Epidemic Modelling

Andrzej Krajka*
Paweł Pawelec

## 1 Introduction

In [10], there is posed the problem of the best model for simulation SARS-CoV-2 epidemic. According to [5], we can make a remark that this epidemic in Poland consisted of different waves, which probably can be associated with the different viral variants (see [4] and the reference given there). However, it is difficult to split one time series data (number of infected or recovered) into the waves connected with the different virus mutations. Therefore, in [3] it is proposed to divide the time interval into periods in such a way that in each period the number of infected people is increasing fast, decreasing fast, or is almost constant. The authors conclude that, based on the USA Covid-19 daily infection data, the best is piecewise SEIUR model. Their investigation has the following weaknesses:

- History of SARS-CoV-2 epidemic in the USA is not typical. The USA is a big country and epidemic waves are the results of interference of different waves in different parts of the USA (*the east coast of the USA, California, Alaska* etc.).
- They compare SEIUR and piecewise SEIUR model with segments chosen manually, only. There is no rule for the divide whole period on segments.

In this paper, we give a dividing method of the whole period into segments, for each country (from among the available data) we automatically divide the whole segment on periods and compare four methods (on whole data and on segments), choosing the best (with respect to the standard error computed from the model number of infected and recovered individuals).

## 2 Models

We consider in this paper four popular epidemic models in the literature.

*Corresponding author — andrzej.krajka@poczta.umcs.pl

**SIR**:

$$\begin{aligned} \frac{dS(t)}{dt} &= -\beta I(t)S(t), \\ \frac{dI(t)}{dt} &= \beta I(t)S(t) - \gamma I(t), \\ \frac{dR(t)}{dt} &= \gamma I(t), t \geq 0, \end{aligned} \tag{1}$$

which divides (in each moment $t$) the whole population into those suspected of being ill ($S(t)$), those who are currently infected ($I(t)$), and those who are recovered (with immunity, $R(t)$).

However, for the most infectious diseases, there is a latent period between being infected and becoming infectious: the exposed group ($E(t)$). We extract this group in this and the following two models.

**SEIR**:

$$\begin{aligned} \frac{dS(t)}{dt} &= -\beta I(t)S(t), \\ \frac{dE(t)}{dt} &= \beta I(t)S(t) - \sigma E(t), \\ \frac{dI(t)}{dt} &= \sigma E(t) - \gamma I(t), \\ \frac{dR(t)}{dt} &= \gamma I(t), t \geq 0. \end{aligned} \tag{2}$$

**SEIR2**:

$$\begin{aligned} \frac{dS(t)}{dt} &= \sigma - \sigma S(t) - \beta I(t)S(t), \\ \frac{dE(t)}{dt} &= \beta I(t)S(t) - \sigma E(t), \\ \frac{dI(t)}{dt} &= E(t) - (\gamma + \sigma)I(t), \\ \frac{dR(t)}{dt} &= \gamma I(t) - \sigma R(t), t \geq 0. \end{aligned} \tag{3}$$

**SEIUR**:

$$\begin{aligned} \frac{dS(t)}{dt} &= -\beta S(t)(E(t) + U(t)), \\ \frac{dE(t)}{dt} &= \beta S(t)(E(t) + U(t)) - \sigma E(t), \\ \frac{dI(t)}{dt} &= \sigma f E(t) - \gamma I(t), \\ \frac{dU(t)}{dt} &= \sigma(1 - f)E(t) - \gamma U(t), \\ \frac{dR(t)}{dt} &= \gamma(I(t) + U(t)), t \geq 0. \end{aligned} \tag{4}$$

where in these models we introduce the fifth group of people named unreported infected $U(t)$.

We are to minimalize the value

$$Err = \sum_{t \in T} [(\hat{I}(t) - I(t))^2 + (\hat{R}(t) - R(t))^2],$$

where $\hat{I}, \hat{R}$ are the observed values of the infected and recovered fraction of people, $I, R$ are values obtained from the model, and $T$ is the time interval.

## 3 Piecewise models

Looking at Figure 1 in [5], we see that the values $\beta$ and $\gamma$ in the SIR model are drastically changed in time. This figure was created with the SIR model for the increasing time interval with a fixed left point interval. We see that it is difficult to say of one epidemic of SARS-CoV-2. The situation in Poland can be characterized as different "waves" of epidemics with different $\beta$ and $\gamma$ characteristics. For $t \in [5, 225]$, there are two stationary points $(\beta, \gamma) : (1, 0.946)$ and $(0.524, 0.475)$. These points characterize a very short time of new transmissions and a short time recovery. These values suggest there exists an overlap of two different epidemic processes (probably in different regions of the country). For the time interval $[227, 270]$ all solutions were not stable (*ode* procedure shows *Abnormal termination*). This chaotic behavior ends with two stationary points in the time $[270, 404] \cup [461, 507]$ with $(0.36, 0.31)$ and in the time $[405, 460] \cup [508, 568]$ with $(0.27, 0.23)$. Looking at the behavior of the real infected number $\hat{I}$ in Poland in Figure 1, we can extract five phases of the epidemic:

- **04.03.20–18.09.20** — stable phase (denoted by 0),
- **19.09.20–23.11.20** — growth phase (denoted by 1),
- **24.11.20–21.01.21** — decrease phase (denoted by $-1$),
- **22.01.21–01.04.21** — growth phase (denoted by 1),
- **02.04.21–26.06.21** — decrease phase (denoted by $-1$),
- **27.06.201–02.10.21** — stable phase (denoted by 0),

and two waves of epidemic (it will be explained in details in Section 5.1).

Obviously, each "single" epidemic model for such data fails. This problem was remarked and considered in details in [3]. In conclusion, the modeling should be done in every one of the intervals described above with the condition of continuity of functions at break points only. Thus, we have four additional methods: piecewise SIR (deonted by SIRp), piecewise SEIR (SEIRp), piecewise SEIR2 (SEIR2p) and piecewise SEIUR (SEIURp).

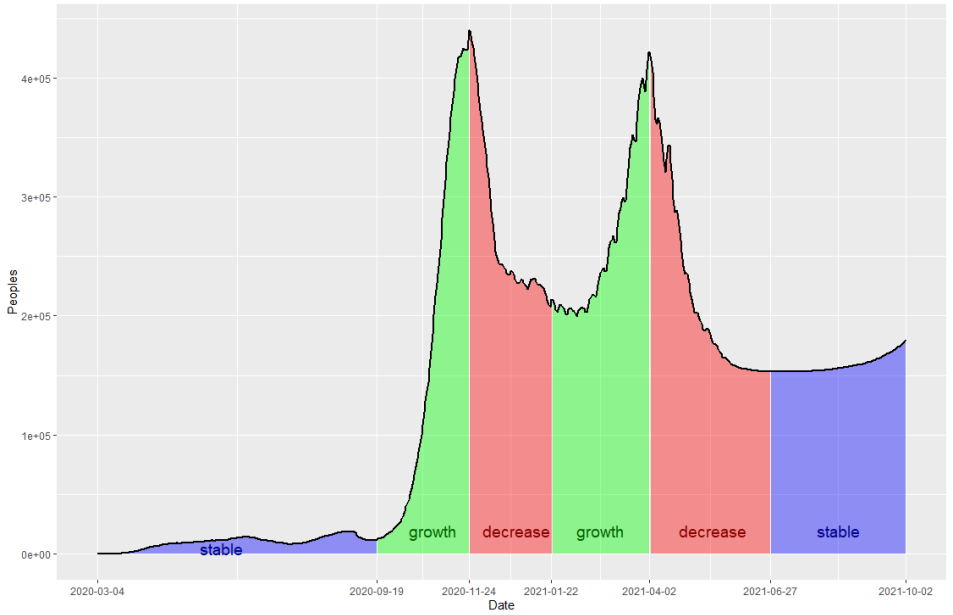



Figure 1: Phases of epidemic SARS-CoV-2 in Poland

# 4 Program

The source data of COVID history ([15]), size of population and population density was pulled out from [11] (in *csv* format), there are a lot of websites containing COVID information ([12, 13, 14]). We read data in the format `country_name, N, confirmed_cum, death_cum, active_cum, recovered_cum, xtime` where `N` is the number of citizens of country, `xtime` is the date of observations, and others are observed values.

For each country, for such prepared data, we make the following steps:

- **Step 1.** We identify the *growth, decrease*, and *stable* phases of time intervals.
- **Step 2.** Using the *optim* function from the *ode* library of R we compute the optimal parameters for each model. We minimize the error function $Err$ given in (2). We repeat this for each piecewise model in such a way that we minimize $Err$, and the starting point of each part is the ending point of previous part.

obtaining the following two data frames:

***wyn*** — with the structure: `country, time, St, It, Rt,` $S_i, E_i, I_i, U_i, R_i, Sp_i, Ep_i, Ip_i, Up_i, Rp_i$, `i=1,2,3,4,` where `St, It, Rt` are observed suspect, infected and recovered; the allowed values are obtained from models:

***wcoeff*** — with the structure: `country, method, from, to, sigma, beta, gamma, sig, f_param, S0, E0, I0, U0, R0,` describing each segment of

approximation (or the whole time interval) with time (`from, to`) parameters and initial values.

Data frame *wcoeff* is the base to the product plot in Figure 2, whereas on the base *wyn* we recompute $Err$ and produce Figures 3, 4, and the Table 2.

**Step 1** is difficult and probably the most interesting part of this program, therefore we explain this step in detail. For the aim of identification of phases of $I$ function we use the $R$ function *ets* for 25 times differentiating (function *diff(Rt, 25)*) recovery observations $Rt$ time series. Now we compute the difference of the smoothing (trend) time series and compare a single increase with the maximum increase of the smoothing time series. When the single increase is smaller then $-0.01\times$ max single increase then this point is marked $-1$, if the single increase is bigger then $0.01\times$ max single increase then this point is marked $+1$, otherwise, we marked it as 0. Multiplying by the maximal single increase is made to create a model fitting regardless of the size of the country (number of people). The second part of this procedure is devoted to the pooling of values $-1, 0, +1$ into the intervals, removing intervals shorter than 15 to obtain vectors of breaking points with names of $-1, 0$, or $+1$. This fragment is presented in Listing 1.

```
smooth <- function(Rt, xlev=0.01, xprog=15, ndiff=25){
  w<-ets(diff(Rt,ndiff))

  # Identyfing the regions growth, decrease and stable
  xd<-diff(w$states[,1],1)
  xds<-rep(0,length(xd))
  xds[xd< -xlev*max(xd)]<- -1
  xds[xd> xlev*max(xd)]<- 1

  # Change the series on breaks segments and values
  xdf<-c(1); xdt<-c(xds[[1]]); l=1
  for (i in 2:length(xds)){
    if (xdt[l]==xds[i]) xdf[l]<-xdf[l]+1 else {
      xdt<-c(xdt,xds[i])
      xdf<-c(xdf,1)
      l<-l+1
    }
  }

 # The segments shorten that xprog are denoting by NA
  xdt[xdf<xprog]<-NA

  # connecting areas from NA to neighbors
  xdf1<-xdf[1]; xdt1<-xdt[1]; l=1
  for (i in 2:length(xdf)){
    if (is.na(xdt1[l]) && is.na(xdt[i])) xdf1[l]<-xdf1[l]+xdf[i] else {
      xdt1<-c(xdt1,xdt[i])
      xdf1<-c(xdf1,xdf[i])
      l<-l+1
    }
  }

  for (i in 1:length(xdf1)){
    if (is.na(xdt1[[i]])){
      if (i==1) xdf1[i+1]<-xdf1[i+1]+xdf1[i]
      if (i==length(xdf1)) xdf1[i-1]<-xdf1[i-1]+xdf1[i]
```

Table 1: Periods of epidemic

| 20-03-04 | 20-09-19 | 20-11-24 | 21-01-22 | 21-04-02 | 21-06-27 | 21-10-02 |
|---|---|---|---|---|---|---|
| 0 | +1 | −1 | +1 | −1 | 0 | |

```
      if (i>1 && i<length(xdf1)) {
        xx<-floor(0.5*xdf1[i])
        xdf1[i-1]<-xdf1[i-1]+xx
        xdf1[i+1]<-xdf1[i+1]+xdf1[i]-xx
      }
    }
  }
  xdf1<-xdf1[!is.na(xdt1)]
  xdt1<-xdt1[!is.na(xdt1)]

  if (length(xdt1)>=2) {
  for (i in 2:length(xdt1)){
    if (xdt1[i-1]==xdt1[i]){
      xdt1[i-1]<-NA
      xdf1[i]<-xdf1[i-1]+xdf1[i]
    }
  }
  }
  xdf1<-xdf1[!is.na(xdt1)]
  xdt1<-xdt1[!is.na(xdt1)]
  xdf1[1]<-xdf1[1]+floor(ndiff/2)
  xdf1[length(xdf1)]<-xdf1[length(xdf1)]+ndiff-floor(ndiff/2)
  names(xdt1)<-xdf1

  return(xdt1)
}
```

Listing 1: The function which indentify segments in SARS-CoV-2 recover observations

The values `xlev=0.01, xprog=15, ndiff=25` were chosen experimentally.

For **Step 2**, we use *optim* from the library *deSolve* to fit the SIR, SEIR, SEIR2, SEIUR models to our data. Some details and tips can be found in the paper [3].

# 5 Results

## 5.1 Waves of epidemics in different countries

From the data frame *wcoeff* we have the areas of growth, decrease, and stability of the epidemic. For data from Figure 1, we have pieces (Table 1) giving the areas of epidemic $\{0, +1, -1, +1, -1, 0\}$. Deleting 0 and dividing the sequence ointo subsequences such that the first number of the subsequence is $+1$ and the subsequence contains at least $-1$ we have two waves for Poland: (from 2020-09-19 to 2021-10-22 and from 2021-10-22 to 2021-06-27). Proceeding in such a way for each country, we obtain the map of waves.

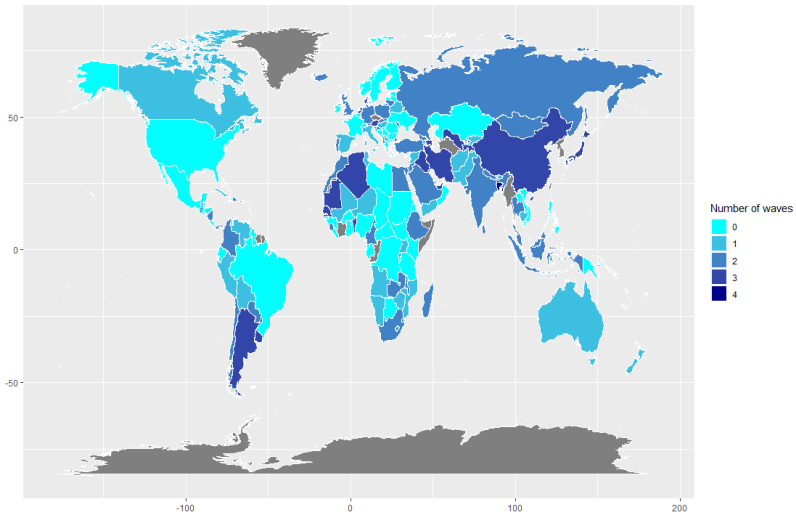



Figure 2: Number waves of epidemic SARS-CoV-2 in different coutries

As previously mentioned, we remark that big countries (except China and Argentina) have a smaller number of waves than the small ones.

## 5.2 Classical SIR model

In this section, we evaluate the error of classical SIR methods computed for different countries. Here, we apply the error

$$MAPE = \sum_{t \in T} \frac{|\hat{I}(t) - I(t)| + |\hat{R}(t) - R(t)|}{\hat{I}(t) + \hat{R}(t)} \tag{5}$$

and show the error in a $\log_{10}$ scale.

The MAPE errors are in most countries extremely big.

## 5.3 The best method of modeling for every country

In Figure 4, we present the best (in the sense of minimizing $Err$ from 2) method of SARS-CoV-2 modeling for each country. Additionally, we compute the number of countries with the best method.

Moreover, we calculate the Pearson correlation coefficients for all the methods (piecewise or not) with the number of waves obtained 0.4995.

# 6 Conclusions

We stress the conclusions in the points:

- It is not true that the piecewise SEIUR method is the best for each country (21.9% only), the piecewise SIR is better (25.3%).

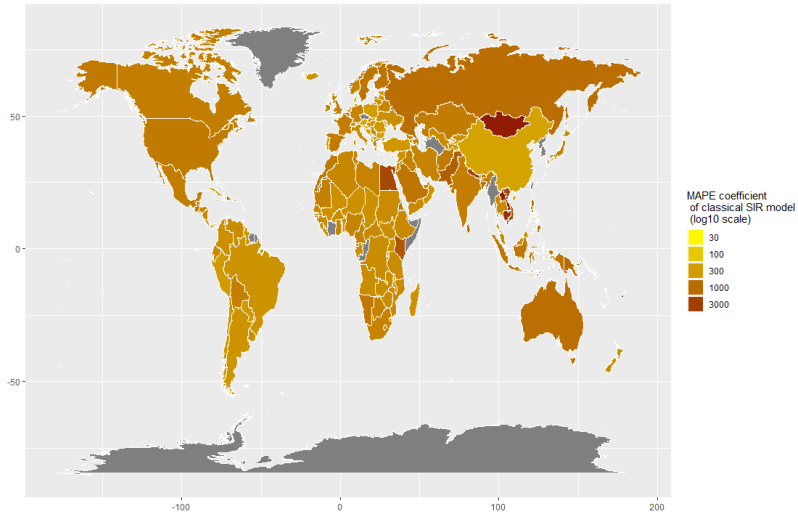



Figure 3: MAPE error of SIR method simulation of SARS-CoV-2 epidemic

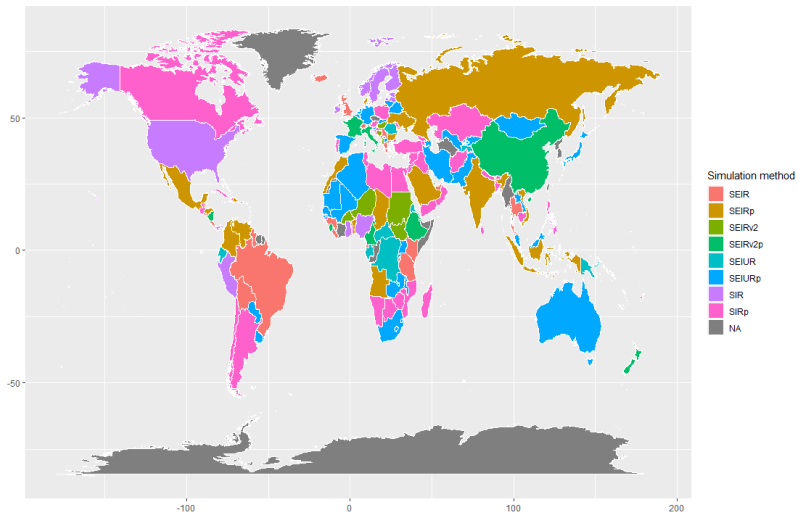



Figure 4: The best simulation method for SARS-CoV-2 epidemic

Table 2: Percent of countries with the best modeling method

| Method | Full | Piecewise | Together |
|---|---|---|---|
| SIR | 9.89 | 25.27 | 35.16 |
| SEIR | 18.13 | 9.89 | 28.02 |
| SEIR2 | 4.40 | 6.04 | 10.44 |
| SEIUR | 4.40 | 21.98 | 26.37 |
| Together | 36.81 | 63.19 | 100.00 |

- SEIR methods given by (2) formula are significantly better than those given by (3).
- Piecewise methods are almost 2 times better than the one for the whole observation interval. As it was predicted, the big influence on this result has a big number of waves (correlation almost 0.5), but not only.
- Non-high position of SEIUR and SEIR methods can come from difficulties with evaluation of the number of exposed people $E$ which is not observed and must be approximated.

# References

[1] Allen, L. J. (2010). An introduction to stochastic processes with applications to biology. CRC press. https://sistemas.fciencias.unam.mx/~silo/Cursos/coronavirus/Allen.pdf

[2] Andersson, H. & Britton, T. (2012). Stochastic epidemic models and their statistical analysis (Vol. 151). Springer Science & Business Media. http://archive.schools.cimpa.info/archivesecoles/20160119113752/t_britton.pdf

[3] Chen, Z., Feng, L., Lay Jr, H. A., Furati, K., & Khaliq, A. (2022). SEIR model with unreported infected population and dynamic parameters for the spread of COVID-19. *Mathematics and computers in simulation,* 198, 31-46.

[4] Cosar B, Karagulleoglu ZY, Unal S, Ince AT, Uncuoglu DB, Tuncer G, Kilinc BR, Ozkan YE, Ozkoc HC, Demir IN, Eker A, Karagoz F, Simsek SY, Yasar B, Pala M, Demir A, Atak IN, Mendi AH, Bengi VU, Cengiz Seval G, Gunes Altuntas E, Kilic P, Demir-Dora D. SARS-CoV-2 Mutations and their Viral Variants. Cytokine Growth Factor Rev. 2022 Feb;63:10-22. doi: 10.1016/j.cytogfr.2021.06.001. Epub 2021 Jul 2. PMID: 34580015; PMCID: PMC8252702.

[5] Kamińska Sandra, Krajka Andrzej Antoni, SARS-CoV-2 Epidemic in Poland and Other Countries In: Selected Topics in Applied Computer Science, Vol.II, 2023 / Bylina Jarosław Marcin, Wójcik Grzegorz, 2023, vol. II, Lublin, Uniwersytet Marii Curie-Skłodowskiej w Lublinie, s.47-58, ISBN 978-83-227-9675-7

[6] Kermack W. O. and McKendrick A. G., Contributions to the mathematical theory of epidemics–I., 1927, *Bulletin of mathematical biology*, 53, 1-2, p. 33–55, 1991.

[7] Liu, Z., Magal, P., Seydi, O., & Webb, G. (2020). A model to predict COVID-19 epidemics with applications to South Korea, Italy, and Spain. *medRxiv*, 2020-04.

[8] Ritchie H., Mathieu E., Rodés-Guirao L., Appel C., Giattino Ch., Ortiz-Ospina E., Hasell J., Macdonald B., Beltekian D. and Roser M., (2020) - "Coronavirus Pandemic (COVID-19)". Published online at OurWorldInData.org. Retrieved from: https://ourworldindata.org/coronavirus [Online Resource]

[9] Wu, Y., Sun, Y., & Lin, M. (2022). SQEIR: An epidemic virus spread analysis and prediction model. *Computers and Electrical Engineering,* 102, 108230.

[10] Yang, H. M., Lombardi Junior, L. P., & Yang, A. C. (2020). Are the SIR and SEIR models suitable to estimate the basic reproduction number for the CoViD-19 epidemic?. *MedRxiv*, 2020-10.

[11] https://raw.githubusercontent.com/RamiKrispin/coronavirus/master/csv/coronavirus.csv

[12] https://data.europa.eu/euodp/en/data/dataset/covid-19-coronavirus-data

[13] https://data.europa.eu/euodp/en/data/dataset?vocab_concepts_eurovoc=http%3A%2F%2Feurovoc.europa.eu%2F838

[14] https://github.com/owid/covid-19-data

[15] https://ourworldindata.org/covid-stringency-index

# A Generalization of Lotka-Volterra Model

Andrzej Krajka*
Marcin Szyszka

## 1 Introduction

The Lotka-Volterra model described by two populations (numbers of predators $P$ and preys $V$), firstly formulated in [4] and [9], can be written as follows

$$\begin{aligned} \frac{dV(t)}{dt} &= rV(t) - aV(t)P(t), \\ \frac{dP(t)}{dt} &= -sP(t) + abV(t)P(t)), \end{aligned} \tag{1}$$

where $V$ — number of preys, $P$ — number of predators, $r$ — prey per capita growth rate, $a$ — the effect of the presence of predators on the prey growth rate, $ab$ — the predator's per capita death rate, and $s$ — the effect of the presence of preys on the predator's growth rate.

The model (1) has a lot of generalizations (cf. [6, 7] or [2]). Full history of these generalizations can be found in [3]. It is worthwhile to remark that as shown in [8], the Lotka-Volterra model is increasingly unstable with the greater than 2 number of populations. In this paper we propose some generalization of Lotka-Volterra model to more than 2 populations with food preferences described by the graph. We emphasize the stability of the proposed model as well as the stability of the described ecosystem. Here, we present one example of an ecosystem and observe the change in the quantity of populations when one population is deleted. Another generalization of Lotka-Volterra law can be found in cf. [5]. All our computations are done with the R language (version 4.3.1) on the RStudio platform (version 2023.06.2 561).

## 2 Generalization of Lotka-Volterra model

The graph (Figure 1) describing the food preferences of organisms (for two populations $P$ and $L$ will be denoted by $w_{P->L}$). The large weight of an edge indicates

*Corresponding author — andrzej.krajka@poczta.umcs.pl

that this organism especially likes to eat the second one. Lack of edges indicates that these two organisms are not in the predator-prey role. For each population $L$, we give the initial quantity of population $N_o(L)$, maximal environmental capacity of this population $N_{max}(L)$, and fraction $hidden(L), 0 < hidden(L) < 1$ of organisms which are 'hidden' in each step of simulation, i.e. which do not 'participate' in a predator-prey interaction as prey. Two constants $\gamma_1$ and $\gamma_2$ characterize the degree of 'predators' and 'preys', respectively. We choose these constants and fix its values experimentally, but in a lot of applications it is possible to choose them according to the population $\gamma_1(L), \gamma_2(L)$.

If the population $L$ is both predator (for organisms $V_1, V_2, ...V_k$) and prey (for organism $P_1, P_2, ...P_m$) then we can describe our model as:

$$\begin{aligned}\frac{dL(t)}{dt} &= s_L(1 - L(t)/N_{max}(L)) \\ &\quad -\gamma_1 * L(t) * (1 - hidden(L)) * \sum_{P:P->L} P(t)w_{P->L} \\ &\quad +\gamma_2 * L(t) * \sum_{V:L->V} (1 - hidden(V))V(t)w_{L->V}.\end{aligned} \tag{2}$$

The function $f_1(x) = 1 - x, x = L(t)/N_{max}(L)$ describes the 'free' area for expansion of the population for $x < 1$ and the shrinking area for $x > 1$ therefore, this function should be positive for $0 < x < 1$ and negative for $x > 1$. However, it is true only for $s > 0$. If $s < 0$, the function $f_1$ should be always positive for $x > 0$ and significantly greater for $x > 1$. In this paper, we do not consider the case of negative $s$. But the linear behavior of this function in the neighborhood of $x = 1$, is not satisfying in a lot of applications. Therefore, we will consider the functions:

$$\begin{aligned}f_2(x) &= max\{min\{-log(x), 1\}, 0\}, \\ f_3(x) &= 1 - x^2, \\ f_4(x) &= 1 - \sqrt{|x|}, \\ f_5(x) &= \begin{cases} 1 - x, & \text{if } x \leq 1, \\ -\exp|1 - x|, & \text{otherwise} \end{cases}\end{aligned}$$

Furthermore, in the Lotka-Volterra model, as well as in (2), it is assumed that all organisms in the population is ready to breed, which in many populations is not realistic. Therefore, we introduce the delay of breed as the functions $g_i, i = 1, 2, 3,$

$$\begin{aligned}g_1(x) &= 0, \\ g_2(x) &= rgeom(1, prob = 0.5), \\ g_3(x) &= rpois(1, 0.8),\end{aligned}$$

where $rgeom(1, prob = 0.5)$ denotes the random number chosen from geometric law with parameter $p = 0.5$ and $rpois(1, 0.8)$ denotes the random number chosen

Table 1: Graph properties of $\mathcal{G}$

| Attribute | Value |
|---|---|
| Assortativity | -0.61735 |
| Mean betweenness | 0.80000 |
| Diameter | 2.00000 |
| Mean harmonic centrality | 8.56071 |
| Transitivity | 0.16363 |

from the Poisson law with the parameter $\lambda = 0.8$. Now the proposed model (2) should be rewritten:

$$\begin{aligned} \frac{dL(t)}{dt} &= s_L f(L(t-1-g(t))/N_{max}(L)) \\ &\quad -\gamma_1 * L(t) * (1 - hidden(L)) * \sum_{P:P->L} P(t) w_{P->L} \\ &\quad +\gamma_2 * L(t) * \sum_{V:L->V} (1 - hidden(V)) V(t) w_{L->V}, \end{aligned} \tag{3}$$

with $f = f_i, 1 \leq i \leq 5$ and $g = g_i, i = 1, 2, 3$. The general aim of this paper is to establish:

- What is the best choice of $\gamma_i, i = 1, 2$, values of model (3) to establish stability of this model?
- What is the influence of the choice of functions $f$ on the simulation run?
- What is the influence of the choice of functions $g$ on the simulation run?
- How stable is the ecosystem when one of the organisms is deleted (one of the populations dies)?

## 3 Data

Our interspecies relation is illustrated by the graph $\mathcal{G}$ in Figure 1. We will denote by $\omega = \omega(\mathcal{G})$ the adjacency matrix of graph $\mathcal{G}$ .

The graph properties of $\mathcal{G}$ are listed in Table 1.

The remaining parameters taken in the simulation are summarized in Table 2 and equation (4).

$$\begin{aligned} N_o(L) &= \frac{N_{max}(L)}{2}, \\ hidden(L) &= 0.01, \\ \gamma_1 &= \alpha_1 * \min_L(|s(L)|), \\ \gamma_2 &= \alpha_2 * \min_L(|s(L)|), \end{aligned} \tag{4}$$

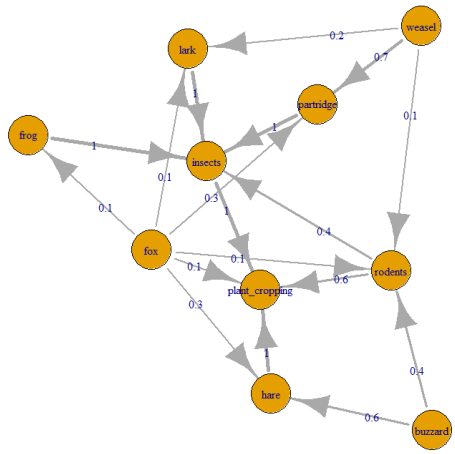



Figure 1: Example ecosystem graph $\mathcal{G}$

Table 2: Parameters of simulation for the graph $\mathcal{G}$

| Population (L) | $s$ | $N_{max}$ |
|---|---|---|
| weasel | 0.01 | 260 |
| buzzard | 0.01 | 100 |
| fox | 0.01 | 50 |
| lark | 0.03 | 100 |
| partridge | 0.06 | 300 |
| frog | 0.04 | 200 |
| insects | 0.07 | 130 |
| rodents | 0.05 | 250 |
| hare | 0.02 | 300 |
| plant cropping | 0.03 | 100 |

In the next section we show the details of the program, in Section 5 we will consider the stability of simulation with respect to different pairs $\alpha_1, \alpha_2$, whereas in Section 6 we will consider the influence of the lack of one of described above factors.

# 4 Program details

If $w$ denotes the adjacent matrix of $\mathcal{G}$ and $Hidden$ is the vector of hidden fractions of populations, we prepare values:

Algorithm 1: **Preparation of initial parameters**

```
ns<-length(Hidden)
w1 <- apply(w,1,sum)     # sum _{V} w(L->V)
w2 < -apply(w,2,sum)     # sum _{P} w(P->L)
w1[w1==0] <- 1           #  in order to not dividing by 0
w2[w2==0] <- 1           # in order to not dividing by 0
wK <-  (t(Hidden) %*% w)[1,]
```

The following fragment of the listing is the kernel of the simulation. We compute the vector of numbers of individuals of populations $il$ at the moment $t$ on the basis of numbers of individuals of populations $il$ in the time $t-1$ under given functions $f, g$ and values $\gamma_1, \gamma_2$.

Algorithm 2: **The main loop of simulation**

```
ilosc[1,]<-il<-N0
for (t in 2:Tmax){
   wi1 <- (w %*% il)[,1]      # sum_{P} w(P->L) il(L)   L,1,2,3,...,
   wi2 <- (t(il) %*% w)[1,]   # sum_{V} w(L->V) il(L)   L=1,2,3,...
   tau <- pmax(t-1-g(ns),1)   # ns - the number of populations
   x <- rep(0,ns)
   for (i in 1:ns) x[i] <- ilosc[tau[i],i]
   il <- s*il*f(x/Nmax)-gamma1*il*(1-Hidden)*wi1/w1+
            gamma2*il*(wi2-wK)/w2+il
    il[il < 0] <-  0
   ilosc[t.] <- il
}
```

# 5 Results

## 5.1 Stability of the model

We tested the model behaviour with respect to $\alpha_1$ and $\alpha_2$ coefficients. As the test we computed for each pair of $(\alpha_i, i = 1, 2)$ generalized Lotka-Volterra model $(T_{max} = 300)$ matrix with the numbers of all organisms (columns) in each moment $t = 1, 2, 3, \ldots T_{max}$. Algorithm 2 contains the matrix `ilosc`. Then we compute $\omega = \max_L\{\max_{1 \le t \le T_{max}} L(t)N_{max}(L)\}$. If the model 'works' correctly, this value should not be greater than 10. The value $\beta = \log_{10}(\omega)$ for different values $\log_{10}(\alpha_1)$ and $\log_{10}(\alpha_2)$ on the $OX$ and $OY$ axis, respectively, is shown in Figure 2.

It seems that acceptable values of $\beta$ should belong to the $(-\infty, 0]$ interval (then no population does not exceed its maximal capacity). Hovewer, due to not

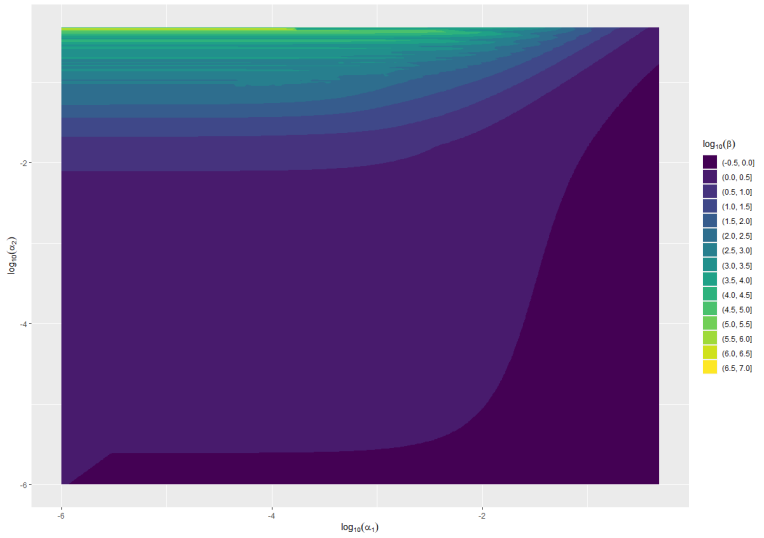



Figure 2: Stability of proposed Lotka-Volterra model

always realistic assumptions on the initial state of population, until the numbers of individuals of all populations stabilize, the exceeding of maximal capacity of population size is very likely. We see this on the example run in Figure 3. Therefore, we regard the values $\beta \in [-1, 1]$ as the stability area.

## 5.2 The influence of the choice of functions $f$ and $g$ on the simulation process

The example run of the simulation for the graph $\mathcal{G}$ described in Figure 1 with parameters $\alpha_1 = 0.1, \alpha_2 = 0.05, T_{max} = 300, f = f_3, g = g_2$, we present on the Figure 3

The change of $g_2$ on $g_1$ or $g_3$ did not influence the run of simulation, it slightly shifts on the axis $OX$, only. Whereas the change of function $f$ has a significant influence on the number of individuals of populations that are close to the maximal capacity, and in consequence on other populations, too. Because, in our simulations on Figure 3 the population *plant cropping* is near the maximal capacity, we present the results of five simulations of graph $\mathcal{G}$ with the functions $f = f_i, i = 1, 2, 3, 4, 5$ and other parameters as in Figure 3 and in Figure 4 show the run of numbers of individuals for *plant cropping*, only. We see that the least restrictive is function $f_2$ and the most restrictive is function $f_5$ which 'not allows' to exceed the max capacity level ($y = 1$). The reverseback of another functions is weak and allows for maintaining the population size above $1.1 \times max\ capacity$ over the long period of simulation.

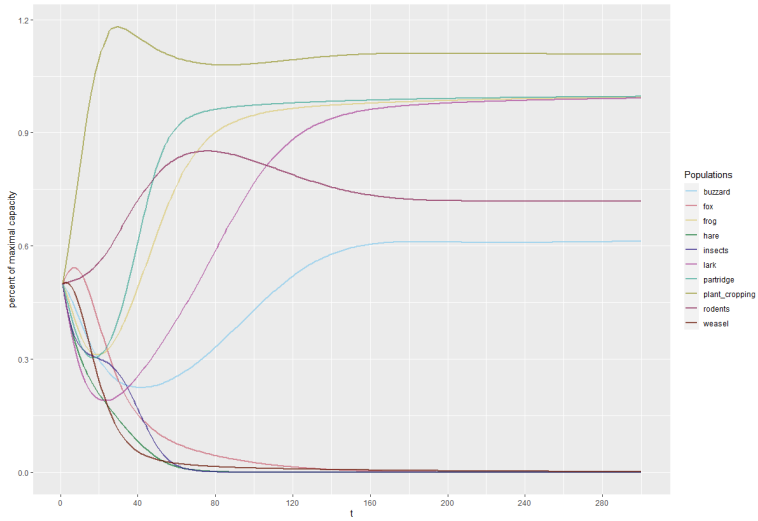



Figure 3: Example of simulation of Lotka-Volterra model

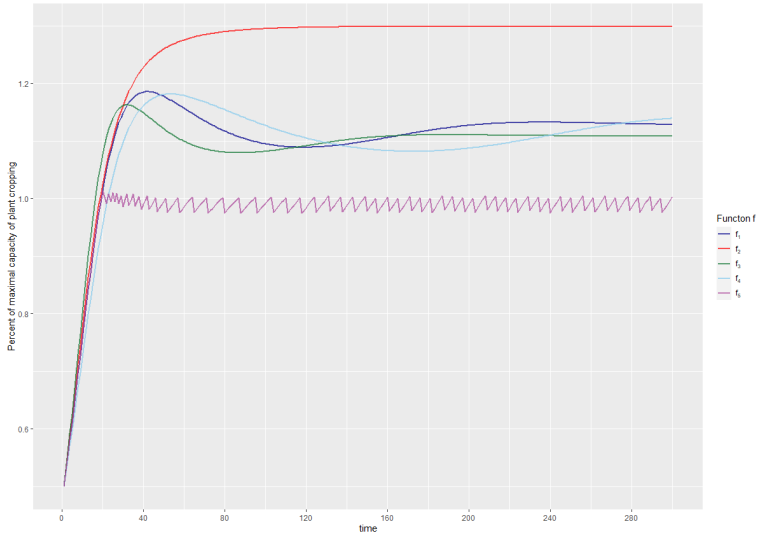



Figure 4: The quantity of plant cropping for different functions $f$

Table 3: Table of benefits after deleting the population

| Deleted | weasel | buzzard | fox | lark | partridge | frog | insects | rodents | hare | plant cropping |
|---|---|---|---|---|---|---|---|---|---|---|
| weasel | 0 | **0.00** | 0.99 | 1.00 | 1.00 | 1.00 | 1.05 | 0.99 | 1.02 | 1.11 |
| buzzard | 0.89 | 0 | 0.49 | 1.00 | 1.00 | 1.00 | 0.87 | 0.95 | 0.86 | 0.99 |
| fox | 0.98 | 1.00 | 0 | 1.00 | 1.00 | 1.00 | 1.04 | 0.99 | 1.02 | 1.01 |
| lark | 51.27 | 0.61 | 1.79 | 0 | 0.83 | 0.58 | 0.02 | 1.19 | 0.42 | 0.99 |
| partridge | 44.56 | 0.21 | 4.04 | **0.00** | 0 | 0.02 | **0.00** | 1.33 | **0.00** | 1.00 |
| frog | 80.22 | 0.30 | 2.36 | 0.04 | 0.36 | 0 | 3.9e7 | 1.27 | 1.57 | 0.96 |
| insects | 0.99 | 1.00 | 0.99 | 1.00 | 1.00 | 1.00 | 0 | 0.99 | 0.99 | 1.07 |
| rodents | 0.32 | 2.03 | 12.21 | 1.00 | 1.00 | 1.00 | 0.19 | 0 | 0.18 | 0.99 |
| hare | 0.99 | 1.00 | 1.01 | 1.00 | 1.00 | 1.00 | 1.04 | 0.99 | 0 | 1.09 |
| plant cropping | **0.00** | **0.00** | **0.00** | 1.01 | 1.00 | 1.01 | 2.21 | 1.39 | **0.00** | 0 |

### 5.3 Stability of ecosystems

At the time $t = 100$ of the simulation, we delete one population (say $p$, it dies) and continue the simulation to the time $T_{max} = 300$, obtaining the vector of quantities of individuals of all populations. Dividing this vector by the vector of quantities of individuals of all populations (without deletion population $p$, it is a vector of values from Figure 3 for $t = T_{max}$), we obtain the vector of *benefits* of populations from deletion population $p$. The values greater than 1 indicate that deletion of $p$ affected the increase in quantity of the considered population. In contrast, the values smaller than 1 indicate that the quantity of individuals is smaller than that with $p$ not deleted. We present the *benefits* values in Table 3, such that the name of the deleted population $p$ is in the first row, the *benefits* of populations (names in columns) are in the sequential rows. The populations that are in consequence endangered or dead are highlighted in bold red.

In the ecosystem defined by graph $\mathcal{G}$, *weak links* are *partridge* or *plant cropping*. It is natural, because *plant cropping* is the direct and indirect prey for most organisms. Significant instabilities arise when we delete *frogs* from the ecosystem. The deeper biological conclusions should be taken from the more realistic graph $\mathcal{G}$.

## 6 Conclusions

In this paper, we present the proposition of the generalization of the Lotka-Volterra model on the case of a higher number of populations in which the dependencies are described by the directed graph with weighted edges. Our model is much more stable, providing the appropriate choice of parameters, which characterize the greater stability. We do not observe characteristic oscilations for two populations; in our models, the numbers of individuals of each population are rather stabile. The biological applications of our model are an open question, but the applications seem to be accurate.

## References

[1] Din, Q. (2013). Dynamics of a discrete Lotka-Volterra model. *Advances in Difference Equations,* 2013, 1-13.

[2] Foryś, U., & Poleszczuk, J. (2011).*Modelowanie matematyczne w biologii i medycynie.* Warszawa: Wyd. UW. https://mst.mimuw.edu.pl/wyklady/mbm/wyklad.pdf

[3] Knuuttila, T.,& Loettgers, A. (2017). Modelling as indirect representation? The Lotka–Volterra model revisited. *The British Journal for the Philosophy of Science.* https://www.jstor.org/stable/pdf/26494784.pdf?casa_token=6F3BRrRtmMYAAAAA:PFRECiFnNSv2eSUkZoP08N6D3UOzQsDyjNmVfoUXJmEtAGHzXCooGg9BqsOGQihRqEtOxDf5_Vv7rUWw3GfVyz6uscq-kHEwwjTRwqK9oSX9froUPA

[4] Lotka, A. J. (1956). *Elements of mathematical biology.* Dover Publications.

[5] Mao, X., Sabanis, S., & Renshaw, E. (2003). Asymptotic behaviour of the stochastic Lotka–Volterra model. *Journal of Mathematical Analysis and Applications,* 287(1), 141-156.

[6] Murray, J. D. (1989). Mathematical biology, vol. 19 of *Biomathematics.*

[7] Murray, J. D., & Murray, J. D. (2003). *Mathematical Biology: II: Spatial Models and Biomedical Applications* (Vol. 3). New York: Springer. https://ds.amu.edu.et/xmlui/bitstream/handle/123456789/7253/Mathematical%20Biology%20I.%20An%20Introduction%20Third%20Edition%20-%20J.D..pdf?sequence=1&isAllowed=y

[8] Vano, J. A., Wildenberg, J. C., Anderson, M. B., Noel, J. K.,& Sprott, J. C. (2006). Chaos in low-dimensional Lotka–Volterra models of competition. *Nonlinearity,* 19(10), 2391. https://sprott.physics.wisc.edu/pubs/paper288.pdf

[9] Volterra, V. (1931). Variations and fluctuations of the number of individuals in animal species living together. *Animal ecology*, 412-433.

# Author index

The creation of this book was inspired by the growing complexity and interdisciplinary nature of modern computational research, spanning fields such as linguistics, neuroscience, artificial intelligence, and epidemic modeling. In a world where technology continues to integrate itself into our lives and societies, understanding its foundations and practical applications has never been more critical. This volume brings together contributions from diverse research domains, showcasing innovative methods, tools, and theoretical advancements that address contemporary challenges in computation and its applications.

One of the key motivations for this collection was to bridge the gap between theory and practice. Each chapter provides not only theoretical insights but also delves into the application of these ideas in real-world contexts. For instance, the development of Picto, a software system for studying language emergence, demonstrates how experimental linguistics can benefit from computational tools to explore communication dynamics. Similarly, contributions focused on EEG signal analysis illustrate how machine learning can enhance our understanding of the human brain and improve applications in medicine and psychology.

We believe this book will capture the interest of anyone intrigued by the continuous evolution of computer science, particularly students and researchers eager to explore its myriad applications.

UMCS
WYDAWNICTWO

wydawnictwo.umcs.eu

ISBN 978-83-227-9872-0