BABEŞ-BOLYAI UNIVERSITY
FACULTY OF MATHEMATICS AND COMPUTER SCIENCE

Computational Models with Applications in Complex Systems

Summary of the doctoral thesis

PhD student: Florentin BOTA
Supervisor: Prof. Dr. Horia F. POP

2021

Contents of the doctoral thesis

Chapter 1

Introduction

Understanding human nature has been a fascinating subject that challenged philosophers, historians, and scientists since the dawn of time. We generally think of ourselves as having basic needs and wants, yet our interactions are one of the most dynamic processes we have experienced so far [GDDG+03]. Both free will and constraints produce unexpected results that have only increased in complexity over time.

My doctoral thesis *"Computational Models with Applications in Complex Systems"* aims to make contributions to this domain, with a particular emphasis on human behavior modeling in complex systems such as the economy. Additionally, contributions are made in the areas of education, multi-agent networks, and machine learning.

In [CCC+18], Calder et al. describes computational models as tools that can help us translate observations into forecasts for future events, experiment with theories, and derive meaning from the results.

Simply stated, in computational modeling, mathematical models are used to simulate and analyze complex systems with the aid of computer science [TTT96].

A system can be defined as a group of interacting or interdependent items that form a unified whole. In this context, a system is defined by its boundaries, internal structure and function or purpose [Bac00, Boc10].

Such systems, which can be natural or man-made, are the focus of interdisciplinary study of *systems theory*. Complex systems present specific properties such as *emergence, non-linearity and self-organization*. The term is also used to describe the study of such structures, their interactions and collective behavior.

Complex systems represent a subset thought to be inherently difficult to model and composed of many interacting elements [Har87]. Human economies, the Earth's global climate, the Internet [Par05], organisms, and the universe itself are all examples of complex systems.

It is necessary to differentiate between complex and complicated. A complicated machine (such as a car engine) is extremely predictable, even though it is difficult to map and comprehend. This is because the components' relationships are predictable.

A complex system is the exact opposite, where insignificant events may have nonlinear consequences. In the following paragraphs, we will provide an outline of the complexity sciences and how our work integrates into the framework.

Rooted on systems science and cybernetics, the study of complex systems evolved over the span of several decades. In 2010, Hiroki Sayama proposed an organizational map that divides complex systems into seven sub-groups: *Game Theory, Collective Behavior, Networks, Evolution and Adaptation, Pattern Formation, Systems Theory and Nonlinear Dynamics.*

This represents the framework where we will start our investigation, an environment where, as Holland noted in 1995, *adaptation itself builds complexity* [Hol96].

Approached problems

For this thesis, we identified several objectives that were pursued during our research:

- Evaluate computational models inspired by nature
- Investigate human learning behavior and social interactions.
- Discover new methods for determining hidden communities.
- Examine the evolution of agents and compare them to existing research.
- Use stochastic algorithms to mimic the unpredictable behavior of humans.
- Analyze the fairness component of the human nature
- Test agent behavior and compare the results with experimental data.
- Develop educational tools that other students may benefit from

One of the first topics we considered was developing a unique alternative to the standard Homo Economicus [MT00, Tha00] model. These agents, referred to as the standard model of economics, make perfect rational decisions, have no impulses, and present an absolute understanding of their environment.

Our proposed model is based on human experience, and several aspects of it can be found in Behavioral Economics [MT00].

We suggest new aspects that could improve or replace the standard economic model, which allow agents to make impulsive, irrational, or socially driven decisions.

Thesis structure

This work is divided into six chapters, which are as follows:

- **Chapter 1**: We introduce the context of our work and the proposed research topics. This section provides an outline for the study of complex systems.
- **Chapter 2**: We present the state-of-the-art on nature-inspired computational models and community structure detection. Additionally, we explore our first project: *VisualAgents*, a framework we used to simulate an ant colony and demonstrate the key properties of a complex system [BD15]. In this chapter, we also discussed related work on human behavior in economic processes and theoretical concepts regarding machine learning.
- **Chapter 3**: This chapter presents our work regarding agent-based computational models, including an adaptive multi-agent system inspired by wasp behavior and our concept of multidimensional economic agents. We were effective in creating economic agents with a realistic distribution of the fairness component, present in many economic processes. These results were published in several papers [SB18, BS21, Bot21b].
- **Chapter 4**: In this chapter, our work on community structure detection is presented. We devised a new method for detecting group structures in multipartite networks that takes into account shadowed relations between nodes that have a shared neighbor. We proposed and validated a fitness feature that takes these relationships into consideration, and can be used to identify populations on synthetic benchmarks as well as in a real-world application, for massive datasets. Additionally, we describe how we used cooperative game theory to identify networks of co-authorship in the scientific writing environment. Our results were published in two papers [GBSL17, GBSL20].
- **Chapter 5**: Throughout this chapter we will explore our proposed deep learning model for software fault prediction that was built using a combination of CK metrics. Then we will present the use of collaborative tools and activities during the Software Systems Verification and Validation (VVSS) course [CCBP21,SB20]. Additionally we will briefly discuss our work in progress regarding managing emotions for teenagers affected by the COVID-19 pandemic.
- **Chapter 6**: This chapter summarizes the major findings of our study and identifies the most relevant directions for future studies.

Original contributions

The primary objective of my thesis is to develop novel computational models and algorithms that can be used for a deeper understanding and prediction of agent behavior in complex real-life processes. Our main contributions are focused on the following directions.

1. From an economics perspective, we designed and implemented a new Multidimensional Economic Agent (MEA).
2. We used Game Theory and genetic algorithms (GA) to detect community structures.
3. From an application perspective, we developed multiple web and 3D frameworks that can be used as educational resources or to aid in mental health recovery.
4. Aditionally, the findings were used to improve educational components related to collaborative learning and code quality.

The contributions are described in Chapters 3, 4 and 5 and were published in eight research papers [GBSL17, SB18, GBSL20, Bot21b, BS21, CCBP21, BD15, SB20]:

- We created an experimental platform to model natural ant colony processes and, more precisely, to show how the Ant Colony Optimization algorithm (ACO) can be used to demonstrate the emergence principle of a complex system. [BD15]. As an application we developed the *VisualAgents framework* which used 3D and Augmented Reality components to simulate a realistic ant colony. This work served as the first step of our investigation.
- We proposed a new wasp-based quest allocation model [SB18] for virtual learning games. The model model uses wasp domination hierarchy to build a unique multiplayer learning environment for the Sotirios video game [Bot12], where each participant has a unique learning curve depending on their performance. We analysed the wasp behavior as a solution for maintaining a balanced multiplayer mode and to maximize team performance inside the game, with good results.
- In our studies, we noted the significance of the social dimension and used cooperative game theory to detect co-authorship networks in the academic writing environment. This method can be successfully used to identify prominent cross-disciplinary authors and publishing patterns in the scientific community by calculating the authors' marginal contribution to the collective gain. [GBSL20]

- We developed a new solution for the problem of detecting community structures in multipartite networks by taking into account shadowed links between nodes that have a mutual neighbor. We proposed a fitness function that takes these relations into account and demonstrated that it can be used to classify populations on synthetic benchmarks as well as in a real-world application [GBSL17].

- We studied the temporal discounting phenomena found in the Rational component of our multidimensional economic agents and evaluated existing solutions from the scientific literature. We then proposed a new hybrid approach, where the comparative results (based on experimental data) showed that our method performs better in most cases [Bot21b].

- We presented a new evolutionary algorithm that showed the viability of creating intelligent agents with human-like behavior. We successfully created economic agents with a realistic distribution of the fairness aspect found in many economic processes. Experimental data from both human and synthetic agents was used to test the model and a Neural Network component was added in this context, with promising results. [BS21] As an application we developed two distinct versions of the *Ultimatum Game Simulator*.

- A software fault prediction framework was designed, using Deep learning methods. Our main challenge was to define a neural network model based on a combination of CK software metrics [SB20] and to establish the parameters that enhance the accuracy of fault prediction. We discovered a high-performing set of metrics and obtained positive results for our Machine learning approach.

- The use of collaborative tools and activities for the Software Systems Verification and Validation (VVSS) course design was analyzed [CCBP21]. This strategy was proposed prior to the COVID-19 pandemic and proved to be an effective approach when students were required to engage in the new online environment. Our survey results revealed that students preferred collaborative activities, and their grades have improved significantly.

- A book on software development efficiency was published [Bot21a]. I used the knowledge we gained through our research of human nature to determine the most effective approaches to motivate students when teaching Computer Science.

We are optimistic that our current proposed models are relevant to the domains discussed in this study and that they will contribute to the provision of more accurate and useful knowledge when modeling complex behaviors.

Keywords: computational models, multidimensional agents, complex systems, game theory, genetic algorithms, ultimatum game, machine learning, game based learning

I would like to express my gratitude to my esteemed supervisor, Prof. Dr. Horia F. Pop, for his scientific supervision during my PhD studies and to my wife Alexandra and our daughter Eva, for their continuous support.

My sincere appreciation also goes out to my collaborators: Noémi Gaskó, Mihai Suciu, Rodica Ioana Lung, Dana Simian, Camelia Șerban, Camelia Chisăliţă-Creţu and Andreea Pop.

For my colleagues at CodexWorks: together, we automated complex processes for enterprise applications and I am sincerely grateful for your support.

I extend my thanks to Susan Shapiro, Dr. Sally Baas from Concordia University, St Paul and Nadia Crișan from Princeton University for including my contributions to the *6 Feet Apart project.*

In memory of prof. Dumitru Dan Dumitrescu, who made this research possible.

Chapter 2

Background information

Computational modeling can be defined as the study of complex systems through the application of computer science and other disciplines such as mathematics, physics, and statistics. By applying computer simulation, we can use the output of a computing system C to describe the behavior of another system S under some conditions [Pic07].

In this chapter I will describe the context of such nature-inspired models, including ant colonies, wasp behavior, communities and human behavior in economic processes.

2.1 Nature-inspired computational models

Agent-based models (ABM) [NH11] can be used to study autonomous agents and their interactions. The theoretical origins are found in the so called *Von Neumann machine*, a self-replicating concept proposed in 1940 by John von Neumann [Pes95, NB+66] and the basis of *cellular automata*.

The central theory is that basic laws can produce complex behaviour and that we can understand complex phenomena through the study of multiple interactions [GR13].

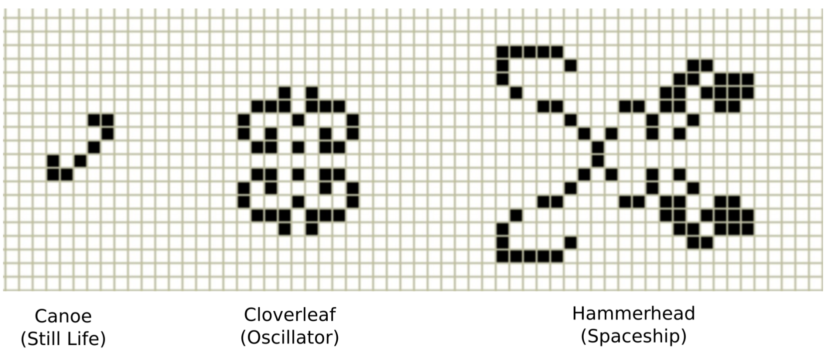


Figure 2.1: Stable patterns in Conway's "Game of Life" [MSJM16]

The first well-known *cellular automaton* is Conway's Game of Life, developed in the late 1960s and based on a simple bi-dimensional array of cells (Figure 2.1).

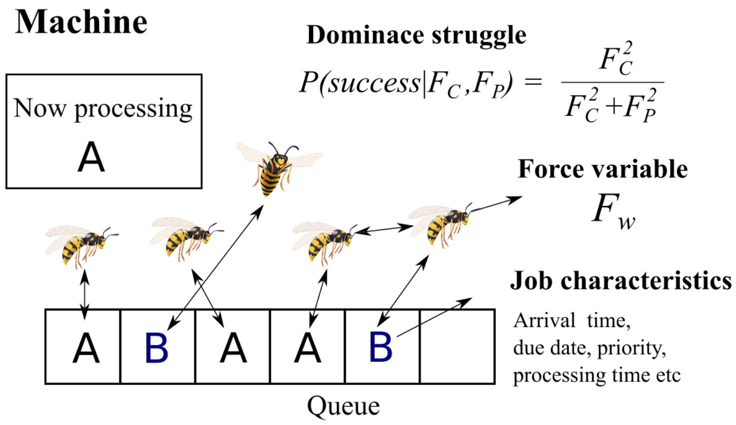


Figure 2.2: Scheduling wasps, Cicirello and Smith, 2001 [CS01]

2.1.1 Models based on wasp behaviour

The model of self-organization in a wasp colony was proposed by Theraulaz et al. in [TBGD91]. The wasp behaviour is characterized by the interaction between the individuals and the environment using a stimulus-response system that controls distributed task allocation.

As mentioned in [SB18], algorithms simulating wasp behavior were used to create multi-agent structures for solving complex problems: self-organizing robots [TBGD91], resource distribution and coordination in a production facility [CS04], distribution of tasks and projects in virtual learning environments [SSMP08], optimization problems [SSS10], dynamic scheduling problems [SM11, Sim09], knowledge mining [DC10], etc. These problems can be related to the task allocation problem in distributed systems proposed by Cicirelo in [CS04]. A dominance contest can arise if two or more wasps compete for the same task with the same probability, as seen in Figure 2.2. The dominance hierarchy is based on a ”force” function that is inversely proportional to the probability of dominance [CS04]. Different models for updating the response thresholds were developed. In [BTD98] the response threshold remain fixed while in [TBD98] is presented the idea of reinforcement of response threshold [SB18].

2.1.2 Ant Colony Optimization and Visual Agents

Marco Dorigo proposed the ant colony system [Dor92] and in this model, the virtual agents are called artificial ants with their behavior being used to find solutions relevant to a given optimization problem. ACO usually reduces the optimization problem into the problem of finding the best route on a weighted graph.

We developed a modern graphical framework for simulating and rendering computational models in 3D. With Unity3D as a graphics engine and C# as the primary

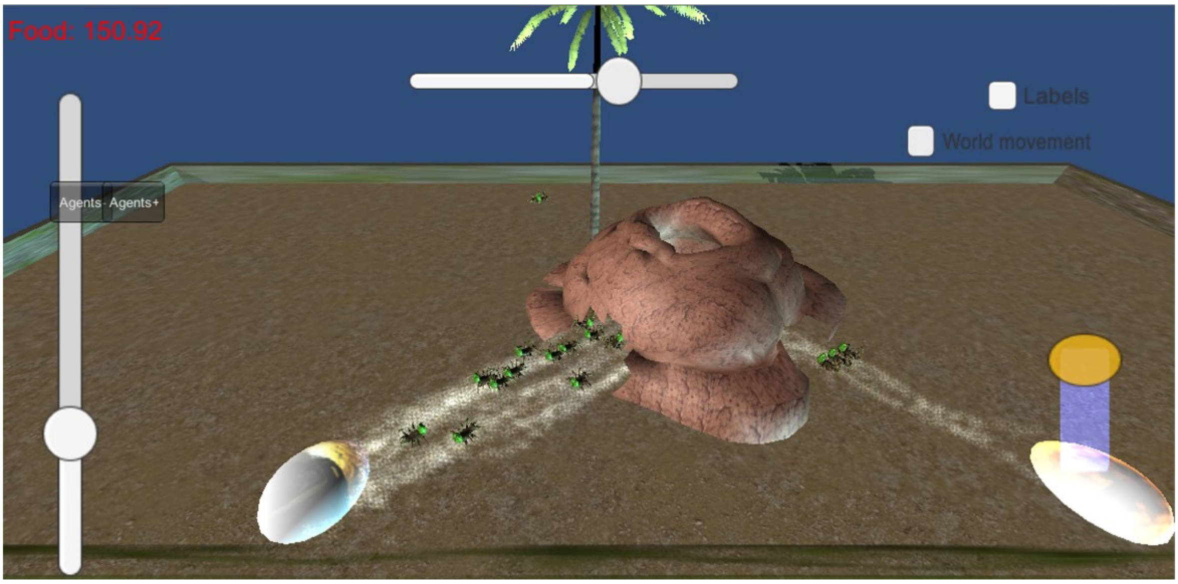


Figure 2.3: Visual Agents - Ant Colony Optimization [BD15]

programming language, we were able to observe processes such as emergent behaviors in a different perspective [BD15]. We started with the ant colony behavior and developed a simulation tool that can be used in future research and for educational purposes. Our main research question in [BD15] was the following.

> How would an ant colony split itself between two sources of food?

As seen in Figure 2.3, we began with a simple environment, a constrained space with a ground layer for navigation and several walls to prevent our agents from falling off the edge during their movement. I used the hive as a central spawn point, where agents can gather to drop their food if they discover it on the map. Users may also adjust their view of the camera, much as in a video game, by using a mouse or, for mobile devices, by touch.

2.2 Human behavior in economic processes

2.2.1 The context

In [F^+12], Farmer argues that we should realistically treat economic systems as complex. His paper starts by comparing the term of *Complexity Economics* with the embarrassment the mathematician Stan Ulam felt in 1977 regarding the title of his lecture "*Nonlinear mathematics*". Ulam stated that since almost all of mathematics and most fascinating phenomena are nonlinear, the term is incorrect and akin to referring to an animal as a "non-elephant animal".

Realistically, businesses are mostly run by human agents who are vulnerable to making errors, mostly due to so-called *cognitive biases*. Cognitive biases [HNM15] are predictable anomalies of reasoning that humans exhibit in their decision. From our observations mentioned in [Bot21b], the standard model is best suited to the macroeconomic system, while the behavioral model is better suited for micro-decisions.

2.2.2 Temporal discounting for economic agents

In this part, I will discuss the background of immediate versus delayed gratification and existing models on how human agents behave when given the option of consuming now versus consuming later. These aspects are based on our research in [Bot21b].

We have already discussed the *utility* concept, which was proposed by Daniel Bernoulli in 1738 and can be used to quantify the satisfaction obtained from the use of a product or service. In this context, **temporal discounting** (or *time preference*) describes the observed phenomenon when agents value the same good (any desired commodity that offers utility to a consumer) *differently* depending on when it is consumed.

The *discount factor*, let's call it δ, is the amount a future value must be multiplied with to obtain the present value and can be calculated using the following formula

$$\delta = \frac{1}{1+\rho} \tag{2.1}$$

where the *discount rate* ρ can be extracted as:

$$\rho = \frac{1-\delta}{\delta} \tag{2.2}$$

This discount rate ρ refers to the discounting rate of interest. We found several examples in the fourth chapter of *Behavioral Economics* by Edward Cartwright [Car18].

2.2.3 Ultimatum Game

To study *systematic errors in judgment*, experimental methods were applied as part of the so-called *"Experimental economics"*. Even more, multiple studies, including [CLS06,BDW03,BdW14,BDGW11,WB20] demonstrated the presence of biases in non-human subjects https://youtu.be/xot4z1CKFMo.

Game theory addresses some of these behaviors by studying strategic interactions between competing players (i.e. games). The mathematics of games have been studied for a long time. For example Blaise Pascal and Christiaan Huygens researched the structure of gambling games in 1957 and Gerolamo Cardano wrote the *"Book on Games of Chance"* in 1564 [Car15]. John von Neumann and Oskar Morgenstern are credited with laying the foundation for modern game theory in 1944 with their analysis on zero-sum and cooperative games [VNM07].

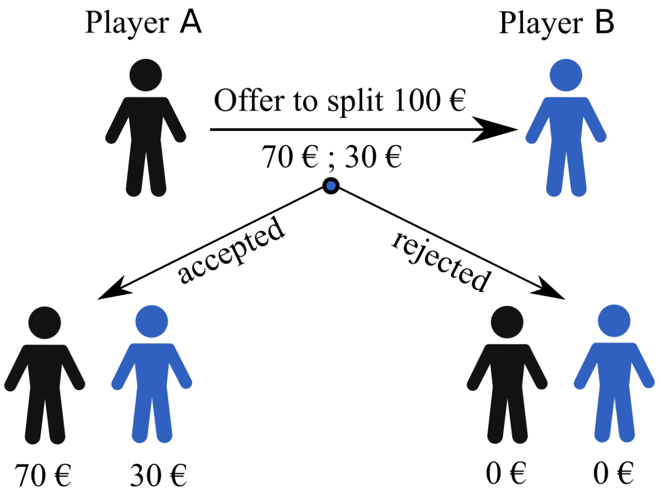


Figure 2.4: Example: Ultimatum Game, amount = 100 EUR

The **Ultimatum Game** (UG) is one of the well-known economics game and serves as a good example of human psychology with apparent non-rational behavior [Gua08, TLS+13] and the rules of the game are simple, as shown in Figure 2.4.

2.3 Community structure detection

Detecting network groups in complex systems is an important challenge, presenting a wide range of applications in fields like criminology (finding potentially dangerous groups), public health (vulnerable groups of individuals), marketing (identifying customer groups), social networks (identifying bot accounts) and many others [KŞ18].

One of the main approaches to community structure detection for unipartite networks transforms the search problem into an optimization problem by using a fitness function that is meant to demonstrate the network's modular structure. One of the most well-known and widely debated fitness functions is the *modularity* [NG04,New06]. The modularity Q_N basically compares a community structure for a network to the corresponding structure when considering a random network. It can be computed as:

$$Q_N = \sum_{C \in \mathcal{C}} \left(\frac{m_C}{m} - \left(\frac{D_C}{2m} \right)^2 \right), \tag{2.3}$$

where m represents the edges, m_C represents the edges in community C, and D_C is calculated as the sum of degrees from all the nodes in C. Normally, a high value for the modularity function means a better solution. In [GBSL17] we investigated this aspect and discovered vary approaches depending on the type of network and solution concept.

2.4 Neural Networks

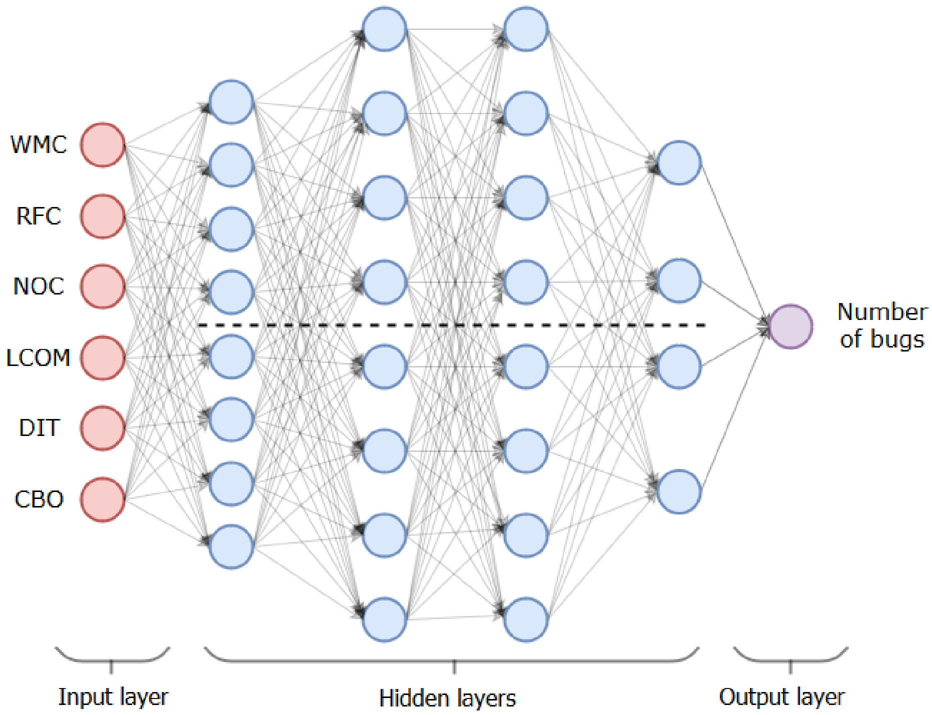


Figure 2.5: Feed-forward Neural Network [SB20]

Artificial Neural Networks (ANN) [RN95] simulate the behavior of the natural brain. They are computing systems made up of interconnected nodes, just like neurons in the human brain and can be organized in specialized layers.

In Figure 2.5 we can see an example of such layered feed-forward neural network (LFFNN) from [SB20], where the first layer is used as an input and the final layer as an output. The nodes between them are part of the so-called “hidden layers”. In this network, a node can only be connected to another node from the next layer and we have multiple hidden layers.

The learning process is straight-forward: the input is given through the first layer and then the output is compared with the expected value. The difference is called the “error” and can be used to adjust the involved weights.

Chapter 3

New Agent-Based Computational Models

Nature presents us with a plethora of representations of both problems and solutions and experts have used these circumstances to advance mankind since the dawn of time. I must admit that observing natural systems is a fascinating activity for me, and I always use my imagination to simulate how they function at both the macro and micro levels. In the following sections I will discuss my work in several projects, where I used artificial agents to simulate specific processes and behaviors.

3.1 Wasp-inspired algorithms in game-based learning: Sotirios

My research began with nature-inspired algorithms, and one of the first systems we studied was the wasp nest. Based on my previous work of creating an educational video-game, I integrated a new model for puzzle allocation based on wasp-like behavior.

Sotirios [Bot12] is an e-learning application (Digital Game Based Learning) which combines cutting-edge graphics with learning puzzles. The narrative of the game revolves around a robot who opposes the planet leaders and the player has to solve different learning puzzles(Figure 3.1) in order to progress in the immersive world of the game. [SB18]

3.1.1 Methodology

My approach on this project was inspired by the human learning behavior. More precisely, the manner in which children and even adults *learn by play.* I have noticed that students get bored or lose concentration during a lecture, but they have no trouble playing video games for hours or even days.

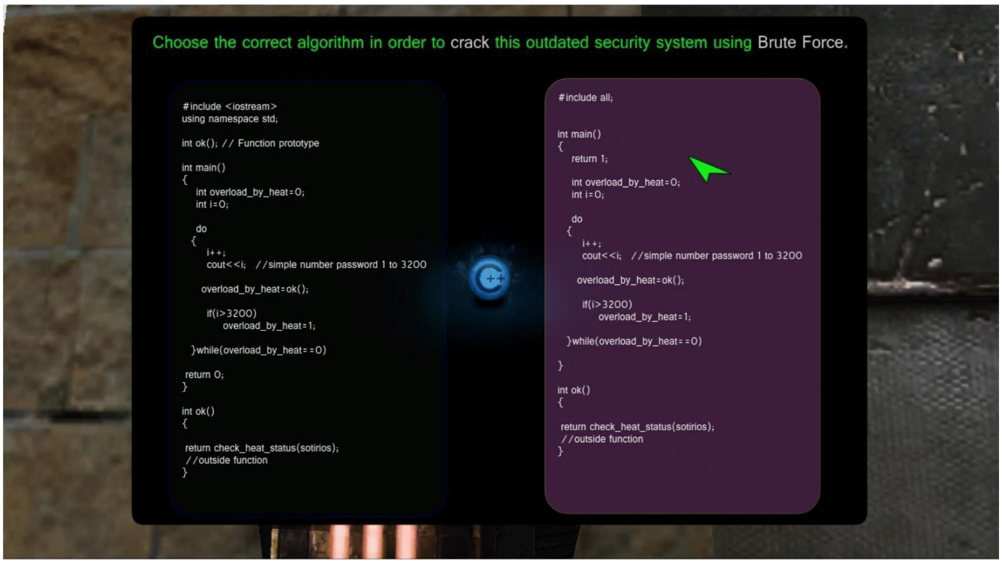


Figure 3.1: User choices for in-game learning, Project Sotirios [Bot12]

3.1.2 The proposed model

In [SB18] we presented our proposed solution for the problem of dynamically distributing puzzles and assignments for multiple players in the game. We addressed the team component in SOTIRIOS to further increase the educational potential of the game, based on several findings on the potential positive effect of peer-group learning [Col80]. Specialization of the player i for solving the puzzle k - $PS_{i,k}$

$$PS_{i,k} = \max\left\{0, \frac{ns_{i,k} \cdot l_{i,k} - nus_{i,k} \cdot l_{i,k}}{ns_{i,k} + nus_{i,k}}\right\} \tag{3.1}$$

Where $ns_{i,k}$ is the number of puzzles having one of the types of the puzzle k, successfully solved before by the player i; $nus_{i,k}$ - the number of puzzles having one of the types of the puzzle k, incorrectly solved before by the player i; $l_{i,k}$ - the level of the puzzle counted in $ns_{i,k}$ and $nus_{i,k}$.

The response thresholds have values between $[w_{min}, w_{max}]$. We denote by $w_{i,k}$ the wasp's response threshold in relation to player i for the puzzle k. Initially all the response thresholds are set on the average value $w_{med} = (w_{min} + w_{max})/2$. The puzzle k broadcasts a stimulus S_k based on the complexity of the puzzle C_k. The wasp agent i bids on the task k if

$$MD_i \leq MAXMT \tag{3.2}$$

with the probability

$$P_{i,k} = \frac{(S_k)^\gamma}{(S_k)^\gamma + (w_{i,j})^\gamma} \tag{3.3}$$

When two wasps i and s enter a dominance contest, the probability of wasp i to

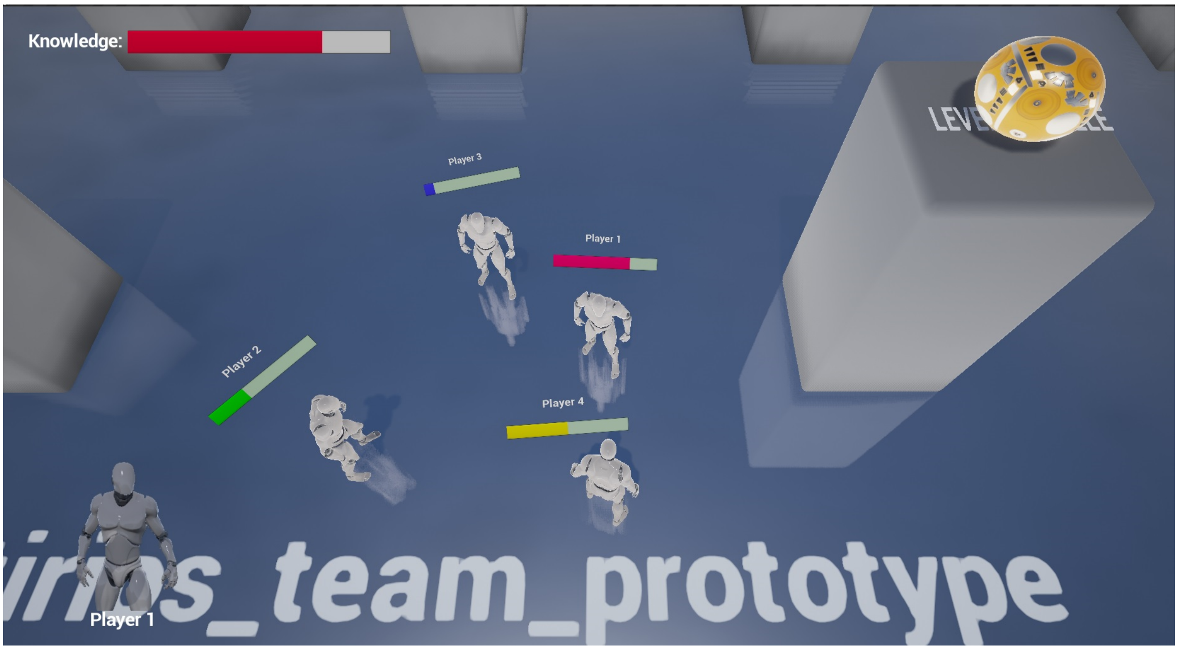


Figure 3.2: Multiplayer version of Sotirios [SB18]

get the puzzle is:

$$P_c(i, s) = P(i\ win|F_i, F_s) = \frac{F_s^2}{F_s^2 + F_i^2} \tag{3.4}$$

3.1.3 Discussion and further directions of study

Learning wasps with allocated to players with similar specialization and equal number of solved puzzles withing the game have equal chances of getting the puzzle.

As mentioned in [SB18], an adaptive reinforcement of the thresholds in the wasp model keeps a balance between the components of the team and avoids the blockage of the game caused by the introduction of a new type of puzzle in the game. Our approach is adaptable and provides for a variety of strategies for the proposed multi-agent system. The parameters can be tuned by hand or other approaches could be developed for this purpose, such as genetic algorithms (GA) or neural networks. Nonetheless, a comparison of several models is necessary to determine the best strategy. Although our alpha tests with undergraduate computer science students yielded encouraging results, more testing and validation are required.

One of the future developments of our work consists in model validation and analysis with different values for the parameters and strategies in the model. Another direction would be to implement the learning structures in other virtual games and compare the results. In Figure 3.2 we can see the in-game visuals of Sotorios, developed using the Unreal Engine.

3.2 Temporal discounting for multidimensional economic agents

In this section, I will discuss my insights on time preference in economic processes. We validated the standard models using empirical data from other related studies and developed our own experiments [Bot21b]. I also compared the models' predictive performance and suggested new hybrid approaches, which were validated. The inspiration for this issue comes from my interest in understanding why people act in certain ways and sometimes make seemingly irrational decisions in a financial context.

We proposed a new unified computational model that reflects human decision-making in complex systems like the economy. The new model is designed using a data-driven methodology [Bot21b] and can be used as an effective tool for developing realistic Multidimensional Economic Agents (MEA). In this project we focused on the *rational* component of the model (the standard paradigm), particularly the time-preference or temporal discounting phenomena that can be seen in the economy.

3.2.1 Methodology

We conducted a survey with students from Babes-Bolyai University of Cluj-Napoca and other participants from an on-line community (reddit) and we used Google Forms as a survey platform.

Table 3.1: Sample survey answers, [Bot21b]

				50$	**5000$**				
3 mo	6 mo	1 yr	2yr	4yr	3 mo	6 mo	1 yr	2yr	4yr
200	400	600	1000	2000	8000	10000	15000	20000	35000
75	100	300	500	1000	5999	7420	8499	9999	14000
60	75	100	200	400	6000	7500	8000	15000	25000
65	80	100	150	300	6000	6666	8000	10000	12500
100	300	1000	2000	5000	10000	10000	12000	15000	20000

The template was:

You are indifferent to Y$ now vs X $ in t years. Write the X amount below

and we questioned the subjects to state indifference for receiving money over 6 months, 1 year, 2 years and 4 years periods. The money amount varied ($50 and $5000).

As compared to older experimental evidence discussed in our work, the findings show some variations, but they are consistent with more recent research [EMT+18].

Proposed functions

In this context, we suggest two new functions that can be thought of as hybrids of existing, simplified models. We believe that by employing multiple parameters, they can more closely approximate modern human behavior, especially given the anomalies we identified in the literature and our own experiments.

Equation 3.5 describes **Hybrid Exponential-Hyperbolic** discount function:

$$hyb_{EH}(x) = \frac{\delta^x}{1 + (\alpha \times x))} \tag{3.5}$$

and equation 3.6 describes the **Hybrid Quasi-Exponential-Hyperbolic** discount function.

$$hyb_{QEH}(x) = \beta \times \frac{\delta^x}{1 + (\alpha \times x))} \tag{3.6}$$

In these functions, δ is the *discount factor* and α represents the discounting parameter that scales the degree of desire for instant gratification, with β being used to capture present-time bias [Bot21b].

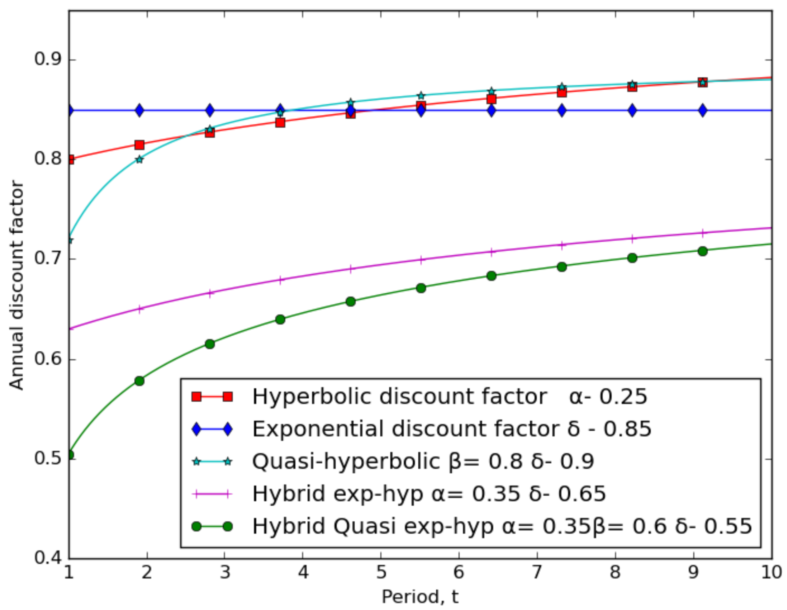


Figure 3.3: Annual discount factor, function comparison, [Bot21b]

Figure 3.3 shows a comparison of current approaches and our suggested hybrid functions for the annual discount factor. We can observe similarities between the quasi-exponential-hyperbolic function and quasi-hyperbolic, and between exponential-hyperbolic with the hyperbolic discount function. As shown in the following section, the variations are important in our analysis.

3.2.2 Experimental Results

The outcome shows a **95.17**% accuracy for the $50 discount and **91.15**% for the $5000 questions. A sample of the answers from our survey in [Bot21b] can be seen in Table 3.1. We tested all multiple functions on our survey data and the results can be observed in Table 3.2.

Table 3.2: Accuracy results for our experimental data [Bot21b]

Discount function	ACC
Hybrid Quasi-Exponential-Hyperbolic	99.39 %
Hybrid Exponential-Hyperbolic	98.65 %
Quasi-Hyperbolic	96.24 %
Hyperbolic	96.07 %
Exponential	89.02 %

3.2.3 Conclusions and next steps

In conclusion, in [Bot21b] we provided an analysis of the temporal discounting phenomena found in economic processes. Using studies from the literature, we evaluated existing discount functions and proposed new hybrid solutions based on our own experimental data.

Our proposed functions performed well and we determined very good results for the Quasi-Exponential-Hyperbolic function, with up to **99.39%** accuracy.

According to our findings, the standard functions performed well on older data sets and struggled to model behavior on newer data, while our proposed solutions show a very good potential to model consumers constraining their own future choices [Bot21b].

In addition, we found promising results with zero-shot and GPT-3 learning in the form of inter-temporal preference (always accepting the present value), with a more in-depth study proposed as the next phase of our research.

3.3 Embedding human behavior using multidimensional economic agents

This section describes contributions to the design of realistic autonomous agents which can simulate the human behavior and respond to multiple factors such as impulse, emotions, social influences and is based on [BS21].

In [BS21] we proposed a stochastic evolutionary algorithm for the development of MEA in a dynamic environment that takes into consideration fairness behavior and community influence as social characteristics. I investigated our agents' evolving perspectives and behavior in an interactive environment.

In our research we examined the one-shot Ultimatum Game and the proposed representation of the agent strategy. In the next paragraphs I will provide our initial results, using the state-of-the-art functions. To ensure that our model is properly validated, we ran our own experiment with human volunteers and corroborated the state-of-the-art empirical results [BS21]. The next sections describe our proposed evolutionary algorithm.

3.3.1 Methodology

In [BS21] we discussed our aim to design and implement artificial Multidimensional Economic Agents (MEA), based on a computational model able to overcome the disadvantages of the models mentioned above and to emulate and predict real human behavior in a way that is useful to both micro and macroeconomics. We propose four components for the computational model that correlate to four levels or dimensions of MEA: *Cognitive, Social, Emotional* and *Rational*

Each component must have a certain weight (significance in the final choice), bias (the possibility of accelerating a decision by itself – raising or reducing the weight by a specific magnitude), and a computing function.

In this investigation, I concentrated on the examination and assessment of the following research questions:

RQ1: Can the perceived human behavior be simulated using evolutionary algorithms?

RQ2: How can we integrate neural networks in our MEA concept?

3.3.2 Proposed Model

Our proposed theory is that the existing human behavior is actually the result of evolution over multiple generations, in a dynamic environment, favoring the emergence of long-term economic winning strategies. These strategies are closely related to human social features, such as fairness and sharing as well as related to environmental restrictions [BS21].

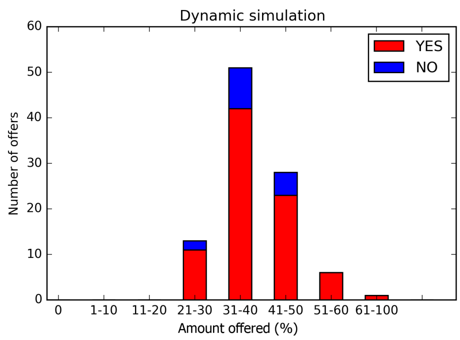
As part of the levels for MEA enumerated before, we proposed a solution for modeling the social and cognitive behavior of human actors who play the economics experiment called "Ultimatum Game" (UG) [Cam11, RTON13]. The UG was used in our MEA study because it provides insights into agent interactions and the consequences of fair and unjust choices. The development of agents capable of reproducing the results of experiments with human subjects while playing UG can occur only when the social dimension of the computational model is taken into account.

3.3.3 Experimental Results

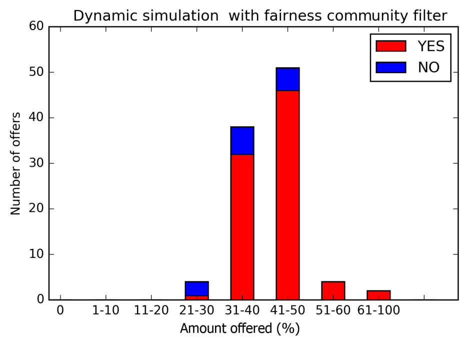
Ultimatum Game Evolution Algorithm

The results obtained in [BS21] using our algorithm proved that natural evolution favors fairness, but the distribution is very dependent on the stochastic parameters.

A further community filter was needed to negotiate offers between the individuals of the same community. In these experiments we used a single community, to better simulate most of our real participants (colleagues).



(a) Simple stochastic dynamic population



(b) Population with the community filter

Figure 3.4: Final results of our agents, [BS21]

Ultimatum Game Simulator

Based on the previous work [BD15] we also implemented several 3D applications, to better illustrate of the concepts involved in the current project. Our first simulation was based on the simplified version of the ultimatum game, where 2 agents play the game as specified in the rules.

Some emotional responses were designed and integrated in our artificial players, as seen in Figure 3.5. In this version, the first player gets random amounts of money in

Figure 3.5: Ultimatum Game Simulator, version 0.7, [BS21]

the $0 - 100$ range and a *confidence* parameter is calculated based on previous games. If the *confidence* factor is over 0.5 (maximum value is 1 and minimum value is 0), the first player makes the offer to the second player [BS21].

Using Neural Networks to Play the Ultimatum Game

In order to simulate the Ultimatum Game, we created a simple feed-forward neural network with the following structure: one node on the input layer, two hidden layers (with 16 and 32 nodes) and a single node on the output layer. We used the ReLU activation function [NH10], *ADAM* optimizer and *mean squared error* as a loss function. The network correctly determined the 40% range based on the training data and offered a limited amount of distribution.

3.3.4 Conclusions

In [BS21] we proposed an evolutionary method to show that intelligent agents with human-like behavior may be created and we validated the algorithm using many experiments with both human and synthetic agents. The results (Figure 3.4) prove that our algorithm allows us to model the human behavior.

In our model, the dynamic environment (changing amounts of money over time during evolution) resulted in a more realistic distribution of fairness. We were able to simulate the results from the real experiment with the help of stochastic evolutionary game theory (Figure 3.4) [BS21]. We also created 3D simulation applications which can be used to describe the concepts involved in this study. The UG Simulator can be accessed on https://codexworks.com/dev/ug/.

The neural network experiment yielded encouraging results in estimating the average offer, and once we address the restricted distribution problem, we can incorporate the component into our MEA model.

Chapter 4

Community Structures: New fitness criteria

4.1 Community structure detection in multipartite networks: a new fitness measure

In [GBSL17] we approached the network community structure detection problem which consists in identifying groups of nodes in a network that are more connected to each other (denser) than to other nodes [For10] and it has been extensively studied for unipartite and bipartite networks.

However, there are very few methods and results extending to multipartite networks, in spite of the fact that numerous applications might benefit from such methods as many living systems or phenomena can be modeled as multipartite networks [LLMW16, AT05]. We proposed an extension of the community fitness defined in [Lan09] for identifying communities in multipartite networks by considering that nodes that have a common neighbor are also connected to each other and thus creating *shadowed* links in the network helping in the discovery of the community structure [GBSL17].

4.1.1 Methodology

The specificity of the multipartite network requires taking into account that, while nodes belonging to the same partition may not be directly connected to each other, they may be indirectly connected by common neighbors in other partitions. Such induced connections may also be responsible for the underlying community structure and should be taken into account when evaluating potential solutions.

In this context we proposed the Multipartite Community Fitness M by extending the *community fitness* in [Lan09] to include such links [GBSL17]. Since these links cannot be considered 'real' network links (if that were the case they would have been

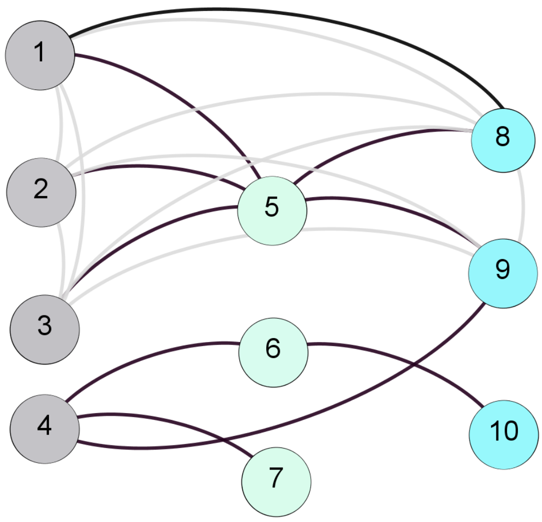


Figure 4.1: A simple tripartite network. [GBSL17]

added at the construction of the network) we call them *shadowed* links and add them to the node degree by using a *multipartite sensibility factor* α.

The fitness of community C is computed as:

$$f_\alpha(C) = \frac{\sum_{i\in C} k_\alpha(i|C)}{\sum_{i\in C} d_{i,\alpha}}, \tag{4.1}$$

as the ratio of the modified total inner degree of the nodes in community C and the total degree of nodes in C. A higher fitness value can be considered to indicate a better community. Considering the same community $C = \{1, 2, 3, 5\}$ in Figure 4.1, its fitness $f_{0.05}(C) = \frac{1+1+1+3.15}{2.05+1+1+5.5} = 0.64$.

The fitness of a *community structure* $\mathcal{C}$, i.e. a partition set over the set of nodes, is computed as the average of the fitnesses of all the communities $C \in \mathcal{C}$:

$$M_\alpha(\mathcal{C}) = \frac{1}{|\mathcal{C}|} \sum_{C\in\mathcal{C}} f_\alpha(C), \tag{4.2}$$

where $|\mathcal{C}|$ is the number of communities.

For weighted networks we use the same approach but considering in all formulas the weighted degree, or weighted inner degree of the nodes, and the sum of the fitnesses of each community in (4.2).

4.1.2 Proposed solution

Maximizing the fitness M should uncover a multipartite community structure in a similar manner a fitness function such as the modularity explores it for unipartite networks. To validate our approach we used an efficient method based on extremal optimization for community structure detection, called *NoisyEO* [ISG17] to maximize the fitness M.

NoisyEO evolves pairs of individuals s and s_{best} by re-assigning random values to the weakest components in s and replacing s_{best} whenever a better solution has been found. s and s_{best} encode possible covers as vectors of size N which represents the size of the network. $s(i)$ represents the community for a node i. The fitness of a node is calculated as the contribution the node has in its community (We measure the community's fitness with and without the given node). Our modification to the original *NoisyEO* algorithm is represented by the new community fitness M in (4.2).

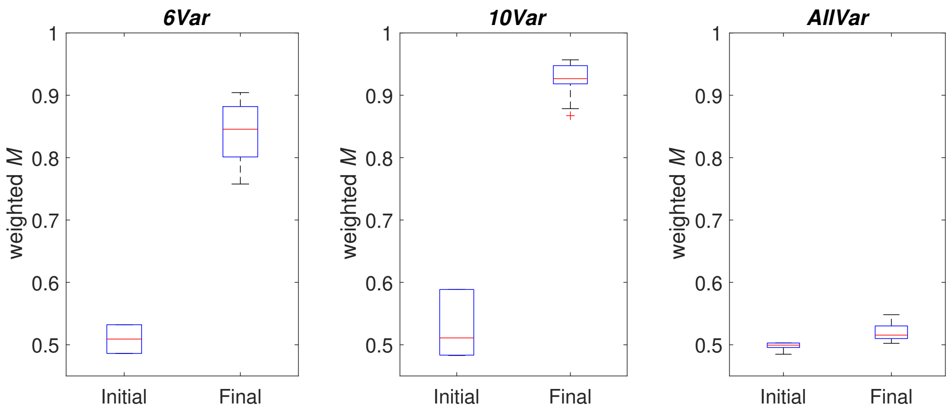


Figure 4.2: Results obtained for the three bank data networks [GBSL17].

4.1.3 Experimental results

Results

The correlation matrix of NMI values of individuals having the best fitness values in the final population shows that the modularity functions are strongly correlated, with 0.92 and similar distributions, while M exhibits a different behavior, but with more NMI values of 1. These graphs show that, while experiments performed on synthetic data do show better results obtained with our multipartite community fitness, there is still room for improvement, as there are situations in which the maximum value of M does not identify the individual having the best NMI value in the population.

Bank marketing data To illustrate the analysis of the bank data we constructed three weighted networks: one containing all variables in the dataset (*AllVar*, 785 nodes, 31549 edges), one that contains only data related to the customer (age-job-marital-education-default-balance-housing-loan-duration-y *10 Var*, 87 nodes, 2845 edges) and one with binary variables removed (age-job-marital-education-balance-duration *6Var*, 79 nodes, 2217 edges) (Figure 4.3).

To interpret results, for example, for the *6Var* network, the algorithm divided people aged below 53 year old from the others. At the job category the algorithm placed in the same community *housmaids*, *retired*, and those with *job-unknown*. By marital status divorced are separated from single and married. Education *primary* is separated form all other levels. This results show how the community structure can reveal connections about categories and help analyze big sets of data.

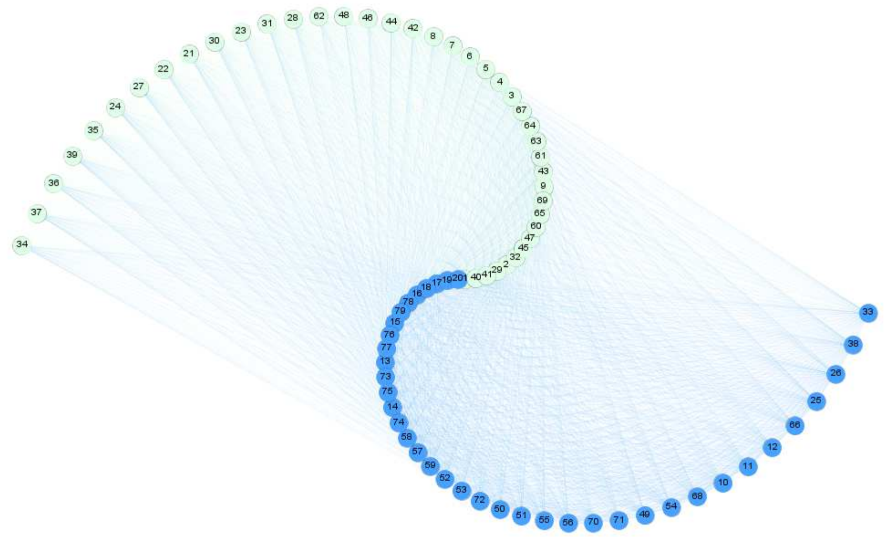


Figure 4.3: The community structure detected in the Bank dataset, *6Var* [GBSL17].

4.1.4 Conclusions

The problem of community structure detection in multipartite networks can be approached by considering *shadowed* connections between nodes having a common neighbor, even if they belong to the same partition.

In [GBSL17] we proposed a fitness function that takes into account these connections and show that it can be used to identify communities on synthetic benchmarks and on a real world application. We can compare results with existing approaches for bipartite networks; numerical results illustrate the efficiency of this approach and identify a new manner of analyzing large datasets.

From a technical perspective, this project provided the most challenging aspects of my research as well as the most satisfying contributions; the full paper was approved and presented at the Genetic and Evolutionary Computation Conference (GECCO).

4.2 A game theoretical analysis of academic writing co-authorship networks

In [GBSL20] we analyzed the field of Academic Writing from a network perspective combined with a game theoretical approach. Co-authorship networks are constructed from publication data from related to academic writing. The Shapley value of each author is calculated, indicating the average marginal collaboration potential of authors. Results offer interesting insights in the publication behaviors of the academic writing community.

The motivation behind our study was as follows. This type of research can provide a snapshot of publications in an emerging field in terms of publishing and may offer students with a much-needed overview of their own field of study. Co-authorship networks have become a Scientometrics analysis staple and their examination may offer important information to a beginner in the discipline.

4.2.1 Methodology

Game theoretical insights

Game theory is an important research field, having an interdisciplinary character with main applications in economics, biology, engineering, politics, etc. Games can be classified into different classes: perfect information & imperfect information games, cooperative vs. noncooperative games, etc.

Shapley value [Har89] defines how important is each player to the coalition, and how the payoff should be split among players.

Network statistics

Because large networks are hard to analyze and compare visually to other networks, network statistics are used to characterize and provide insights into their structure and properties. The modularity measures how well it decomposes into cluster of nodes [BGLL08]. High values indicate complex network structure [GBSL20].

The clustering coefficient [WS98], applied to a single node from the network, measures how complete is the neighborhood of the node, i.e. the proportion between the number of neighboring nodes and the number of total possible neighbors. In a network two nodes are neighbors if there is a link (direct connection) between them. For the network, the clustering coefficient represents the average over all clustering coefficients. It is a value between 0 and 1 and indicates the tendency of nodes to cluster together. An isolated node in the network has value of 0 and a node linked to all other nodes will have a value of 1.

Co-authorship networks and Shapley value

In [GBSL20] we discussed on how network centrality is an important network indicator, with several centrality measures (e.g. betweenness, closeness, PageRank centrality) trying to capture a certain property of the studied networks. In [MAS+13] a linear algorithm is described to obtain the Shapley values of the introduced cooperative game, which represent the average marginal contribution by each node to every coalition of the other nodes. We will use this algorithm to compute the Shapley value of authors that published papers on the topic "academic writing" during 2015 and 2019.

Academic writing co-authorship networks

I collected the data using the Web API of the Scopus database (www.scopus.com) and we filtered the articles that contained the keyword 'academic writing' and were published between 2015 and 2019. We used the following search query:

```
TITLE-ABS-KEY("academic writing") AND ( LIMIT-TO (
DOCTYPE,"ar")) AND ( LIMIT-TO ( PUBYEAR,x) ),
```

where for x we used 5 years, from 2015 to 2019.

We discovered that the networks grow in size throughout the course of the five-year investigation. The first four indications are increasing: the number of nodes, which represents the number of authors who have written in this field, the number of edges, and the average degree, which indicates that successive articles are published in collaboration.

4.2.2 Results

In [GBSL20], the Shapley value for authors in a network is computed as the average marginal contribution to the value of all coalitions of nodes that can be formed with nodes in the network. The value of a coalition is computed as the size of the coalition to which the number of nodes that can be reached at distance 1 from nodes in the coalition.

The Shapley value reveals authors that have the highest marginal contribution to their network – in this case to the publications of a certain year. In the field of academic writing such an information reveals who are the most influential authors in a year. An interesting feature of this field is that papers published in this subject are often aimed at a specific field and such a study would indicate also which fields are formally focusing on the form of presentation of research in articles.

Figure 4.4 presents the co-authorship network for the year 2016 (we have chosen year 2016 because it is the smallest from the five studied networks), without isolated

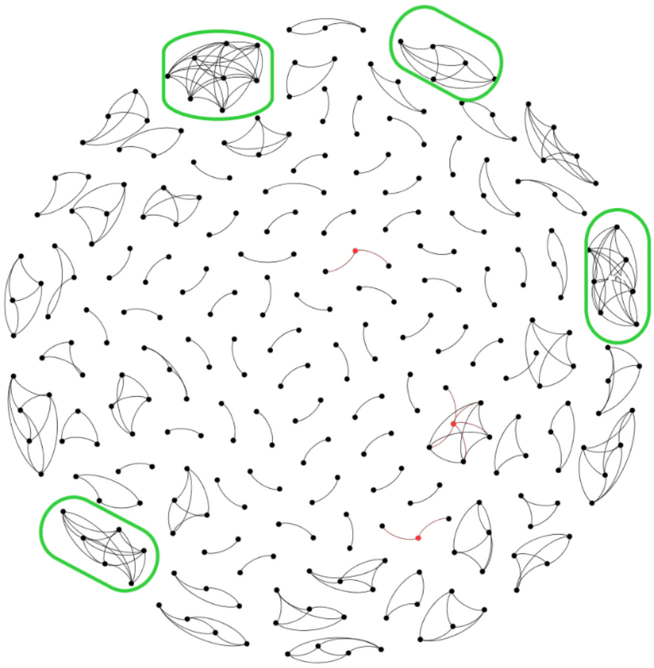


Figure 4.4: The co-authorship network, [GBSL17]

nodes, and highest Shapley values highlighted with red color. We see that the authors with the highest Shapley value are not necessary authors with the highest degrees (connected components marked in green actually reflect one paper with many authors), but those that have contributed more papers with different collaborators, thus potentially contributing to the development of the field.

4.2.3 Conclusions

Academic writing as a research field is a developing research area with particular features, making publication data and trends worth studying. Networks of the academic writing community are analyzed with the help of cooperative game theory.

Shapley value is a solution concept measuring the average marginal contribution of a player to a collective gain. In the case of co-authorship networks it indicates the average potential for collaboration of an author. The most influential authors in the field of academic writing during 2015-2019 are identified for each year. In [GBSL20] we found that their research fields vary indicating a marginal interest to this field while bringing a substantial contribution in the form of journal papers on this topic.

From a technical perspective, using the Shapley value for the investigated author networks and implementing our technique proved to be an engaging approach to the given problem, and I will continue with additional research in the academic writing domain.

Chapter 5

New Educational and Code Quality Aspects

5.1 A conceptual framework for software fault prediction using neural networks

Software testing can be defined as an investigation regarding the quality of the developed product. Finding software defects is a challenging task that usually requires a lot of time and resources. Thus, a mechanism that can be used to predict software faults would provide an advantage in creating better applications with fewer resources.

The contribution in [SB20] is twofold: firstly, we aim to approach the problem of software fault prediction, defining a conceptual framework that incorporates components such as: the object oriented design model, metrics, and methods used for prediction. Secondly, the investigation aims to establish the best-performing combination of CK metrics for predicting software faults.

Metrics Used

Software metrics are very useful when we try to monitor and improve software quality. They are more and more used in Software Engineering domain. Some of the most well-known and studied metrics are those proposed by Chidamber and Kemerer in [CK94]: *Depth of Inheritance Tree* (DIT), *Weighted Methods per Class* (WMC), *Coupling Between Objects* (CBO), *Response for a Class* (RFC), *Lack of Cohesion in Methods* (LCOM), *Number of children (NOC)*

5.1.1 Methodology

In [SB20] we defined the problem of Software Fault Prediction as a regression problem, the goal being to predict the number of faults (bugs) found for a given software component. We created the training set by choosing records from our existing data. We then started the training process with features selected from the CK [CK94] software metrics set. Our goal is to predict, using a neural network model, the number of faults (bugs) that a new class could be affected by.

Objectives and Research Questions

In [SB20] we investigated and validated the following two high level research questions.

RQ1: Does a subset of the CK metrics performs better than all metrics for the task of predicting the number of bugs found in a class?

RQ2: Does the NN based prediction model performs better using a data set defined by one single project than considering the dataset obtained by reunion of all projects?

5.1.2 Proposed model and experiments

Benchmark Dataset

We used the "Bug prediction dataset" from [DLR10], which contains metrics and historical data from several software projects: Java Development Tool (JDT) - Eclipse, Plug-in Development Environment (PDE), Equinox, Lucene and Mylyn. The dataset contains versioning information, the CK metrics, hot-fixes and the number of bugs classified by importance. Mylyn, for example, presents 701 classes with no bugs and 1862 classes with bugs. There were 14577 vulnerabilities in the same project, 235 of which were critical. There are 997 classes in Java Development Tool that have bugs, 432 of which are important. Lucene has no critical or major bugs, but it does have 1714 non-trivial bugs.

Neural Network prediction model description

In order to predict the number of bugs for a given class, a feed-forward neural network with back-propagation learning is used, with the following structure (see Figure 2.5): six nodes on the input layer, one node on the output layer and four hidden layers, each of them having between 64 and 500 nodes. Each node uses the ReLU activation function [NH10]. The termination condition of training is either no improvement of the cross-validation data loss over *100* epochs, or the number of epochs to be at most *1200*. After training this neural network, we obtained a neural network prediction model for the number of bugs in a class.

Experiments description

Our conducted investigation in [SB20] used four experiments, with the five projects/datasets. In each experiment, we trained a neural network-based prediction model using 65% of random sampled data and we used the rest for model validation, a method called *holdout*, the simplest version of cross-validation.

5.1.3 Results and Validation

In [SB20], the initial model did not perform well, with variable accuracy (as low as 0.3 for example) and substantial variations(62.5%). For the second experiment, the discrepancies between training and testing data were insignificant. In the third experiment we found a 42.86% increase in accuracy between the low performing and high performing CK combinations, where (DIT, NOC, LCOM) have 0.35 accuracy, versus (WMC, DIT, LCOM) which present 0.5 accuracy.

We found that there are combinations of metrics *((WMC, DIT, LCOM) and (WMC, RFC, LCOM))* which outperform all the metrics at predicting the number of bugs found in a class. We determined that there is a 21.95% increase in precision by using a high-performing combination of metrics instead of using all the CK metrics. The high performing CK metrics presented a 0.5 precision in our tests, while using all the metrics delivered a 0.41 accuracy. This answers our first research question from section 5.1.1.

An interesting answer for the second question was that when using all the projects from the dataset the predictions performed worse than by using a single project. In our experiments the accuracy was worse by 46.34%, from 0.41 accuracy for the JTD(Eclipse) project to 0.22 for all the projects.

5.1.4 Conclusions and Future Work

In [SB20], we proposed an integrated framework for software fault prediction, defining a neural network based model to empirically establish a combination of CK software metrics that enhance the accuracy for faults prediction. The experimental results showed that there is a 21.95% increase in precision by using a high-performing combination of metrics instead of using all the CK metrics and that a combination of all the projects performed worse in prediction than a single project.

As a future research, we aim to use other machine learning methods for learning or a combination of them, assuming that a hybrid model could be better. Other improvements will be to use other metrics for model validation and different solutions for the problem of imbalanced data.

5.2 Software testing education experiences using collaborative platforms

Collaboration and access to information, according to my observations, are critical when addressing multi-agent systems. This allowed me to propose several aspects of the teaching process and with my colleagues in [CCBP21] I used the opportunity to test the impact team-working and digital tools have on our students' learning processes.

In [CCBP21] we discussed how software testing is a critical step in the software development process. The main contributions in [CCBP21] refer to the following topics: students' perception on learning activities during course length, efficiency improvement while working on subsequent lab assignments, and consequences of adopting online tools in teaching.

5.2.1 Research Methodology

In [CCBP21] we investigated the following research questions:

RQ1: What activities do students find helpful while learning software testing?

RQ2: Can efficiency be improved throughout specific lab assignments?

RQ3: What are the consequences of integrating online tools in VVSS classes?

In order to answer to our research questions we have considered an exit survey filled in by students and the teachers' feedback on the learning activities. The questionnaire had both open and closed questions, asking students' opinion about topics discussed over the course, learning experiences, and tools used to communicate and collaborate in and outside classes. We analyzed 32 complete answers out of 202 students enrolled in VVSS during academic year 2019-2020.

5.2.2 Discussion

RQ1. What activities do students find helpful while learning software testing?

Students consider most relevant for their learning experience those activities that involve interaction and support for practical work. **Live demos** were appreciated by 63% of students as being *very important* in learning software testing. To a higher extent, the **support from teachers** for various activities is perceived as *very important* by 67% of students. The results suggest a group of students focused on problem solving, rather than theoretical aspects. This is confirmed by the 37% and 25% of students that appreciated as *very important* **lectures** and **bibliography**, e.g., papers, reference books on software testing.

RQ2. Can efficiency be improved throughout specific lab assignments?

In the 2019-2020 VVSS course, a total of 202 students created 206 test plans with 1019 test cases in Testlink. Each student was required to implement continuous integration on their projects, either solo or as part of a team. Most of the students chose to work as part of a team, with a number of 79 unique teams being formed. A team could contain a maximum of 3 people, the minimum size of an Agile team [FH+01].

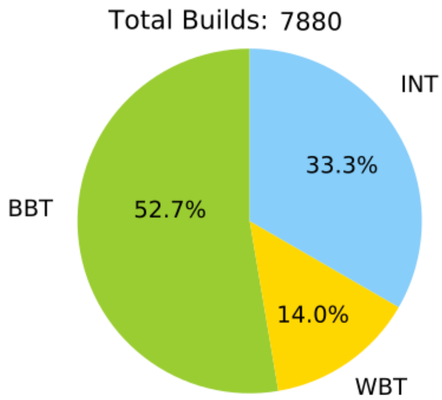


Figure 5.1: Builds distribution exported from Testlink

In figure 5.1 we can observe the distribution of the Jenkins builds per each lab activity. There were a total of 7880 builds, with a large portion on the first lab (BBT) and fewer builds on the next two.

In the first lab of this series (BBT), more builds means the students put more effort to complete a successful test case scenario, perhaps spending more time on bug fixes and environment setup [CCBP21]. The overview of the build status for all the test cases, distributed per labs: BBT had 69.26% passed tests, WBT 76.64% and INT 80.23%, suggesting small improvements throughout lab assignments.

RQ3. What are the consequences of integrating online tools in VVSS classes?

In order to increase quality in teaching software testing, several tools that might ease communication and collaboration were employed for different groups of participants.

The results in Table 5.1 indicate a general increase of grades as Slack tool was employed. During the first three years that were investigated, students worked individually on their assignments. The improved support for students throughout the course activities is reflected by higher grades from year 2018 to 2020.

Table 5.1: Average results from different academic years. [CCBP21]

Year	#Students	Slack	Skype	Labwork in groups	Lab grades mean	Final grades mean
2017	180	no	no	no	5.86	6.88
2018	202	yes	no	no	7.76	7.47
2019	201	yes	no	no	7.71	7.46
2020	202	yes	yes	yes	8.53	8.98
2020	self assessment	yes	yes	yes		7.13

During year 2020 team tasks were assigned for the lab activities in order to improve participants' soft skills. For the lab grades average, we have noticed an increase from 7.46 to 8.98. So far, our analysis on this grade boost is associated to team working activities, since the projects designated to work on during lab activities had similar levels of complexity over all investigated years. Further investigation is required to check whether other factors contributed to this raise [CCBP21].

5.2.3 Conclusions

In [CCBP21] we addressed important aspects related to teaching and learning software testing in current days. Research on this topic indicate the need for a comprehensive approach that manages the learning objectives, the associated activities, tools employed in software testing and the proper training of educators. We have discussed the design of the VVSS course, considering the activities and the employed assessment rules. The efficiency improvement from one lab assignment to another is possible and pushes for skill acquirement. Integration of communication and collaborative tools as Slack and Skype during the course activities improved student grades on formative and summative assessment and helped students gaining essential soft skills.

Future work on the investigated issues aims to deepen the research on increasing testing tool diversity and learning tools during collaborative activities

5.3 The human emotional dimension in the COVID-19 pandemic

This segment presents a new initiative that represents a potential path for our research regarding the human nature.

Our work is done in collaboration with Noah Shapiro, 14 years old, who started the 6 Feet Apart project with the assistance of seven other teenagers and Susan Shapiro, co-author of *The Teacher Within* [SB19]. The project started in early March 2020, shortly after the coronavirus made its way into multiple countries of the world, including the USA. This study is motivated by a need to comprehend the pandemic and to cultivate a sense of community among those affected by the coronavirus outbreak.

Figure 5.2: Virtual Museum, 6 Feet Apart project

Our contributions in this project include the *Stories* web application, the *Book Platform*, and the *Virtual Museum* (Unity 3D) in Figure 5.2. They reflect the project's digital footprint and a technical solution which can help teenagers, teachers and parents to deal with their emotions in the new coronavirus paradigm. The implementation uses Python (Django), Angular and Unity 3D for the Virtual Museum.

This project presents an opportunity to perfect our multidimensional agents and further study the emotional behavior we observed during our research. Based on these results we can decide the appropriate direction of usage for the digital platforms we developed. We hope that the Virtual Museum will remain a historical marker and will act as a repository for the current experiences. The museum can be visited here: http://museum.sharemycovidstory.com

Chapter 6

Final Conclusions and Future Work

Final conclusions

In this thesis we investigated the problem of modeling behaviors in complex systems, with the overall objective of contributing to a better understanding of the problem. We presented state-of-the art solutions and developed new computational models focused on evolutionary algorithms, game theory, and machine learning.

We built an experimental model to demonstrate the natural Ant Colony Optimization mechanism. As a simulation, we created a 3D virtual ant colony and highlighted the emergence concept.

For immersive learning games, we proposed a new wasp-based quest allocation model, which uses a wasp dominance hierarchy to create a one-of-a-kind multiplayer learning experience. We implemented our model in the Sotirios video game and used wasp behaviour to optimize team performance within the game. Introducing gamification strategies to improve learning performance has shown to be a rewarding outcome of our research. Some of these elements were used to refine the VVSS course and evaluate various methods of game based learning.

We used cooperative game theory to detect co-authorship networks in the scientific community, which can help discover cross-disciplinary writers and publishing trends.

Furthermore, we developed a state-of-the art solution for identifying node clusters in multipartite networks. Our approach uncovers multipartite community structures using a new fitness function and extremal optimization *NoisyEO*. Numerical results demonstrate the effectiveness of this technique and establish a novel method of evaluating massive datasets. Our findings have been applied to other fields such as ecology and ethnobiology. [BHBD20].

We developed autonomous agents that can play the ultimatum game with human-like behaviour using a novel evolutionary algorithm in a dynamic environment. The model was validated using experimental data and we introduced a Neural Network feature, with encouraging results. From an application perspective, we developed two variants of the *Ultimatum Game Simulator*.

A system for predicting software faults was developed using deep learning techniques. We proposed a neural network model using a combination of CK software metrics to determine the parameters that improve fault prediction accuracy.

We addressed critical aspects of teaching and learning software testing in the current period and proposed collaboration tools and activities that have been shown to help students improve their technical and soft skills. Additionally, I released a book in this direction, addressing basic problems in soft skills and problem solving that are relevant to the current software development industry.

Future work

We aim to expand our research into the emotional and social dimensions of our agents and model social behavior during chaotic events. Our models may aid in the prediction of economic crises or emotional responses such as those seen during the pandemic or the recent GameStop stock crisis [Bur21].

We discovered new possibilities in the advancement of generative, auto-regressive models from the perspective of machine learning, and we intend to extend similar techniques in other fields, such as psychology and mental health. For example, an interesting approach would be *asymmetric self-play for goal discovery* [OPS+21] or *gated linear networks* [VLB+19], where each neuron tries to predict the output.

Our contributions to the *6 Feet Apart* project will continue, and our work can be of real benefit to children all over the world. The platform will be used in schools from multiple countries, starting with Romania and USA as pilot programs. Teachers and groups of students will express the emotions they experienced during the pandemic and their impressions will be highlighted in the Virtual Museum.

Bibliography

[AT05] Gediminas Adomavicius and Alexander Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE transactions on knowledge and data engineering*, 17(6):734–749, 2005.

[Bac00] Alexander Backlund. The definition of system. *Kybernetes*, 2000.

[BD15] Florentin Bota and Dan Dumitrescu. Agent-based computational models implemented in 3D space. In *Proceedings of Modelling and Development of Intelligent Systems*, pages 13–19, Sibiu, 2015.

[BDGW11] Melissa Bateson, Suzanne Desire, Sarah E Gartside, and Geraldine A Wright. Agitated honeybees exhibit pessimistic cognitive biases. *Current biology*, 21(12):1070–1073, 2011.

[BDW03] Sarah F Brosnan and Frans BM De Waal. Monkeys reject unequal pay. *Nature*, 425(6955):297–299, 2003.

[BdW14] Sarah F Brosnan and Frans BM de Waal. Evolution of responses to (un) fairness. *Science*, 346(6207), 2014.

[BGLL08] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *Journal of statistical mechanics: theory and experiment*, 2008(10):P10008, 2008.

[BHBD20] Avner Bar-Hen, Pierre Barbillon, and Sophie Donnet. Block models for generalized multipartite networks: Applications in ecology and ethnobiology. *Statistical Modelling*, page 1471082X20963254, 2020.

[Boc10] Nino Boccara. *Modeling complex systems*. Springer Science & Business Media, 2010.

[Bot12] Florentin Bota. Game based learning, project Sotirios. In *Proceedings of the second International Students Conference on Informatics*, pages 44–50, Sibiu, Romania, 2012. ICDD 2012.

[Bot21a] Florentin Bota. *Software Booklet: How to Become a 10X Developer*. Colorama Printing House, Cluj-Napoca, Romania, 1st edition, 2021.

[Bot21b] Florentin Bota. Temporal discounting for multidimensional economic agents. *Studia Universitatis Babeș-Bolyai Informatica*, 66(1):86–103, 2021.

[BS21] Florentin Bota and Dana Simian. Embedding human behavior using multidimensional economic agents. In Dana Simian and Laura Florentina Stoica, editors, *Modelling and Development of Intelligent Systems*, pages 3–19, Cham, 2021. Springer International Publishing.

[BTD98] Eric Bonabeau, Guy Theraulaz, and Jean-Louis Deneubourg. Fixed response thresholds and the regulation of division of labor in insect societies. *Bulletin of Mathematical Biology*, 60(4):753–807, 1998.

[Bur21] Riley Burnette. What were the factors that led to the gamestop short squeeze? 2021.

[Cam11] Colin F Camerer. *Behavioral game theory: Experiments in strategic interaction*. Princeton University Press, 2011.

[Car15] Gerolamo Cardano. *The book on games of Chance: the 16th-century treatise on probability*. Courier Dover Publications, 2015.

[Car18] Edward Cartwright. *Behavioral economics*. Routledge, 2018.

[CCBP21] Camelia Chisăliță-Crețu, Florentin Bota, and Andreea-Diana Pop. Software testing education experiences using collaborative platforms. In *Smart Education and e-Learning 2021*, pages 149–159, Singapore, 2021. Springer Singapore.

[CCC+18] Muffy Calder, Claire Craig, Dave Culley, Richard de Cani, Christl A Donnelly, Rowan Douglas, Bruce Edmonds, Jonathon Gascoigne, Nigel Gilbert, Caroline Hargrove, et al. Computational modelling for decision-making: where, why, what, who and how. *Royal Society open science*, 5(6):172096, 2018.

[CK94] Shyam R Chidamber and Chris F Kemerer. A metrics suite for object oriented design. *IEEE Transactions on software engineering*, 20(6):476–493, 1994.

[CLS06] M Keith Chen, Venkat Lakshminarayanan, and Laurie R Santos. How basic are behavioral biases? evidence from capuchin monkey trading behavior. *Journal of political economy*, 114(3):517–537, 2006.

[Col80] Kenneth Gerald Collier. Peer-group learning in higher education: The development of higher order skills. *Studies in Higher Education*, 5(1):55–62, 1980.

[CS01] Vincent A Cicirello and Stephen F Smith. Insect societies and manufacturing. In *The IJCAI-01 Workshop on Artificial Intelligence and Manufacturing, Working Notes*, pages 33–38, 2001.

[CS04] Vincent A Cicirello and Stephen F Smith. Wasp-like agents for distributed factory coordination. *Autonomous Agents and Multi-agent systems*, 8(3):237–266, 2004.

[DC10] Satchidananda Dehuri and Sung-Bae Cho. *Knowledge mining using intelligent agents*, volume 6. World Scientific, 2010.

[DLR10] Marco D'Ambros, Michele Lanza, and Romain Robbes. An extensive comparison of bug prediction approaches. In *2010 7th IEEE Working Conference on Mining Software Repositories (MSR 2010)*, pages 31–41. IEEE, 2010.

[Dor92] Marco Dorigo. Optimization, learning and natural algorithms. *Ph. D. Thesis, Politecnico di Milano*, 1992.

[EMT+18] Kristina Esopo, Daniel Mellow, Catherine Thomas, Hannah Uckat, Justin Abraham, Prachi Jain, Chaning Jang, Nicholas Otis, Michala Riis-Vestergaard, Amanda Starcev, et al. Measuring self-efficacy, executive function, and temporal discounting in kenya. *Behaviour Research and Therapy*, 101:30–45, 2018.

[F+12] J Doyne Farmer et al. Economics needs to treat the economy as a complex system. In *Paper for the INET Conference 'Rethinking Economics and Politics*, volume 14, 2012.

[FH+01] Martin Fowler, Jim Highsmith, et al. The Agile manifesto. *Software Development*, 9(8):28–35, 2001.

[For10] Santo Fortunato. Community detection in graphs. *Physics Reports*, 486(3-5):75–174, feb 2010.

[GBSL17] Noémi Gaskó, Florentin Bota, Mihai Suciu, and Rodica Ioana Lung. Community structure detection in multipartite networks: a new fitness measure. In *Proceedings of the Genetic and Evolutionary Computation Conference*, GECCO '17, pages 259–265, New York, NY, USA, 2017. Association for Computing Machinery.

[GBSL20] Noémi Gaskó, Florentin Bota, Mihai Alexandru Suciu, and Rodica Ioana Lung. A game theoretical analysis of academic writing co-authorship networks. *Journal of Scientometric Research*, 9:319–325, December 2020.

[GDDG+03] Roger Guimera, Leon Danon, Albert Diaz-Guilera, Francesc Giralt, and Alex Arenas. Self-similar community structure in a network of human interactions. *Physical review E*, 68(6):065103, 2003.

[GR13] Volker Grimm and Steven F Railsback. *Individual-based modeling and ecology*. Princeton university press, 2013.

[Gua08] Francesco Guala. Paradigmatic experiments: The ultimatum game from testing to measurement device. *Philosophy of Sciencee*, 75(5):658–669, 2008.

[Har87] David Harel. Statecharts: A visual formalism for complex systems. *Science of computer programming*, 8(3):231–274, 1987.

[Har89] Sergiu Hart. Shapley value. In *Game Theory*, pages 210–216. Springer, 1989.

[HNM15] Martie G Haselton, Daniel Nettle, and Damian R Murray. The evolution of cognitive bias. *The handbook of evolutionary psychology*, pages 1–20, 2015.

[Hol96] John H Holland. *Hidden order: How adaptation builds complexity*. Addison Wesley Longman Publishing Co., Inc., 1996.

[ISG17] Lung Rodica Ioana, Mihai Suciu, and Noémi Gaskó. Noisy extremal optimization. *Soft Computing*, 21(5):1253–1270, 2017.

[KŞ18] Arzum Karataş and Serap Şahin. Application areas of community detection: A review. In *2018 International congress on big data, deep learning and fighting cyber terrorism (IBIGDELFT)*, pages 65–70. Ieee, 2018.

[Lan09] A. Lancichinetti. Detecting the overlapping and hierarchical community structure in complex networks. *New Journal of Physics*, 11:3, 2009.

[LLMW16] Xin Liu, Weichu Liu, Tsuyoshi Murata, and Ken Wakita. Community detection in multi-partite multi-relational networks based on information compression. *New Generation Computing*, 34(1-2):153–176, 2016.

[MAS+13] Tomasz P Michalak, Karthik V Aadithya, Piotr L Szczepanski, Balaraman Ravindran, and Nicholas R Jennings. Efficient computation of the shapley value for game-theoretic network centrality. *Journal of Artificial Intelligence Research*, 46:607–650, 2013.

[MSJM16] Daniel Marolt, Jürgen Scheible, Göran Jerke, and Vinko Marolt. Swarm: a self-organization approach for layout automation in analog ic design. *Int. Journal of Electronics and Electrical Engineering (IJEEE)*, 4(5):374–385, 2016.

[MT00] Sendhil Mullainathan and Richard H Thaler. Behavioral economics. Technical report, National Bureau of Economic Research, 2000.

[NB+66] János Neumann, Arthur W Burks, et al. *Theory of self-reproducing automata*, volume 1102024. University of Illinois press Urbana, 1966.

[New06] Mark EJ Newman. Modularity and community structure in networks. *Proceedings of the national academy of sciences*, 103(23):8577–8582, 2006.

[NG04] Mark EJ Newman and Michelle Girvan. Finding and evaluating community structure in networks. *Physical review E*, 69(2):026113, 2004.

[NH10] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 807–814, 2010.

[NH11] Muaz Niazi and Amir Hussain. Agent-based computing from multi-agent systems to agent-based models: a visual survey. *Scientometrics*, 89(2):479–499, 2011.

[OPS+21] OpenAI OpenAI, Matthias Plappert, Raul Sampedro, Tao Xu, Ilge Akkaya, Vineet Kosaraju, Peter Welinder, Ruben D'Sa, Arthur Petron, Henrique Ponde de Oliveira Pinto, et al. Asymmetric self-play for automatic goal discovery in robotic manipulation. *arXiv preprint arXiv:2101.04882*, 2021.

[Par05] Kihong Park. The Internet as a complex system., 2005.

[Pes95] Umberto Pesavento. An implementation of von neumann's self-reproducing machine. *Artificial Life*, 2(4):337–354, 1995.

[Pic07] Gualtiero Piccinini. Computational modelling vs. computational explanation: Is everything a turing machine, and does it matter to the philosophy of mind? *Australasian Journal of Philosophy*, 85(1):93–115, 2007.

[RN95] Stuart J Russell and Peter Norvig. *Artificial intelligence: a modern approach.* Prentice Hall, New Jersey, 1995.

[RTON13] David G Rand, Corina E Tarnita, Hisashi Ohtsuki, and Martin A Nowak. Evolution of fairness in the one-shot anonymous ultimatum game. *Proceedings of the National Academy of Sciences*, 110(7):2581–2586, 2013.

[SB18] Dana Simian and Florentin Bota. Adaptive multi-agent system based on wasp-like behaviour for the virtual learning game sotirios. In Ivan Lirkov and Svetozar Margenov, editors, *Large-Scale Scientific Computing*, pages 416–424, Cham, 2018. Springer International Publishing.

[SB19] Susan Shapiro and Simona Baciu. *The Teacher Within: A Mindful Journey Toward Well-Being For Teachers In The 21St Century*. Studio Impress Design, 2019.

[SB20] Camelia Serban and Florentin Bota. A conceptual framework for software fault prediction using neural networks. In Dana Simian and Laura Florentina Stoica, editors, *Modelling and Development of Intelligent Systems*, pages 171–186, Cham, 2020. Springer International Publishing.

[Sim09] Dana Simian. Wasp based algorithms and applications. *Proceedings of Int. Conf. Modelling and Development of Intelligent Systems, Sibiu*, pages 229–235, 2009.

[SM11] Marco Santos and Carlos Martinho. Wasp-like scheduling for unit training in real-time strategy games. In *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, volume 6, pages 195–200, 2011.

[SSMP08] Dana Simian, Corina Simian, Ioana Moisil, and Iulian Pah. Computer mediated communication and collaboration in a virtual learning environment based on a multi-agent system with wasp-like behavior. In *International Conference on Large-Scale Scientific Computing*, pages 618–625. Springer, 2008.

[SSS10] Dana Simian, Florin Stoica, and Corina Simian. Optimization of complex svm kernels using a hybrid algorithm based on wasp behaviour. In Ivan Lirkov, Svetozar Margenov, and Jerzy Waśniewski, editors, *Large-Scale Scientific Computing*, pages 361–368, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg.

[TBD98] Guy Theraulaz, Eric Bonabeau, and Jean-Louis Deneubourg. Response threshold reinforcement and division of labour in insect societies. *Proceedings: Biological Sciences*, 265(1393):327–332, 1998.

[TBGD91] G. Theraulaz, E. Bonabeau, J. Gervet, and J. I. Demeubourg. Task differention in policies wasp colonies. a model for self-organizing groups of robots. from animals to animats. In *Proceedings of the First International Conference on Simulation of Adaptiv behavior*, pages 346–355, 1991.

[Tha00] Richard H Thaler. From homo economicus to homo sapiens. *Journal of economic perspectives*, 14(1):133–141, 2000.

[TLS+13] Barbara Tomasino, Lorella Lotto, Michela Sarlo, Claudia Civai, Rino Rumiati, and Raffaella Ida Rumiati. Framing the ultimatum game: the contribution of simulation. *Frontiers in human neuroscience*, 7:337, 2013.

[TTT96] Charles S Taber, Richard J Timpone, and Richard J Timpone. *Computational modeling*, volume 113. Sage, 1996.

[VLB+19] Joel Veness, Tor Lattimore, David Budden, Avishkar Bhoopchand, Christopher Mattern, Agnieszka Grabska-Barwinska, Eren Sezener, Jianan Wang, Peter Toth, Simon Schmitt, et al. Gated linear networks. *arXiv preprint arXiv:1910.01526*, 2019.

[VNM07] John Von Neumann and Oskar Morgenstern. *Theory of games and economic behavior (commemorative edition).* Princeton university press, 2007.

[WB20] Julia Watzek and Sarah Brosnan. Decision-making biases in animals: A critical review. 2020.

[WS98] Duncan J Watts and Steven H Strogatz. Collective dynamics of 'small-world'networks. *nature*, 393(6684):440–442, 1998.