University of Bayreuth

Faculty of Cultural Studies

Bachelor's Thesis

A Game-Theoretical Approach to Understand the Evolution of Theory of Mind

Submitted by:

Tim Birkert

Matriculation Number: 1823100

Study Program:

Philosophy and Economics

Supervisor:

Dr. Paolo Galeazzi

Submission:

23th December 2024 in Bayreuth, Germany

Declaration of Authorship

I hereby declare that this Bachelors thesis, titled A Game-Theoretical Approach to Understand the Evolution of Theory of Mind, is my own work and has not been submitted elsewhere. I have only used the sources and resources stated and cited them appropriately.

T. Birkert

Tim Birkert Bayreuth, 23th December 2024

Contents

Abstract

This thesis examines the evolutionary dynamics of Theory of Mind (ToM). ToM, the ability to attribute mental states to oneself and others, is a fundamental cognitive skill in both cooperative and competitive contexts. Using an agent-based simulation, this study explores the interplay of agents with different ToM capacities across multiple generations to understand why higher-order thinking remains limited in human populations. The Agents compete in dominance-solvable games, with their success determining their reproduction rates. By monitoring these dynamics, the research provides insights into the influence of thinking costs, varying ToM capacities, and alternative strategies on the evolution of Theory of Mind. By integrating insights from evolutionary game theory and cognitive modeling, this work sheds light on the distribution of higher-order thinking in human populations and offers a computational perspective on the interplay between cognitive effort and strategic advantage.

Part I

Introduction

„What does she think I am thinking of her right now?“. Questions like these cross our minds in everyday life. The reasoning about other people's beliefs and mental states is called Theory of Mind (ToM) and a crucial ability to live in a social group and a society. We have to think about the mental states of the people around us to determine socially acceptable actions and to form cooperative bonds. However, the ability to assess the thoughts of others is not only important for communication and cooperation between people but also especially relevant in competitive tasks. This will also be the main focus of this paper.

To illustrate such a competitive ToM task imagine a goalkeeper thinking: „He is probably thinking I will jump in the right corner, therefore he will shoot in the left corner. But what if he knows that I think he is thinking I will jump into the right corner? Then he would assume I rather jump in the left corner and he will shoot in the right corner ... „. Theory of mind is often not a simple but a recursive thinking task, which can go deeper and deeper. It is about correctly assessing the mental states of other individuals to react most preferably. In this example, the goalkeeper tries to asses what the shooter thinks in order to know in which corner he has to jump to prevent a goal.

A plausible assumption could be the deeper one individual can think the more effective it will be in competitive scenarios. As Nagel (1995)[21] shows in her field studies of the Beauty contest shows this is not necessarily the case. This assumption is also implausible due to the fact that we as humans only developed a relatively low depth of higher-order thinking. The distribution of higher-order thinking levels was explored in an experimental study by Kneeland (2015)[16]. The results indicated that 6% of participants were level-0 thinkers, meaning they did not consider the mental states of others at all. Meanwhile, 23% were classified as level-1 thinkers, capable of thinking about what others were thinking, and 27% were level-2 thinkers, who could reason about what others were thinking about their own thoughts. Levels 3 and 4 were each represented by 22%. Similar findings have been reported in other experimental studies. For example, Camerer et al. (2004)[8] analyzed data from 24 beauty contests and found that the distribution of higher-order thinking levels closely followed a Poisson distribution with a lambda value of 1.61. This further supports the observation that most individuals operate at thinking levels between 1 and 3. Similarly, Mauersberger, Nagel, and Bühren (2020)[18] reported comparable results when analyzing the beauty contest data from Bosch-Domenech et al. (2002)[4], showing that most participants adopted strategies consistent with those expected from agents at levels 1, 2, and 3. The main question for this paper is: why do we see this seemingly low distribution of higher order thinking in human populations? Is it solely due to the significant cognitive effort required to perform such complex recursive tasks, or could there also be strategic reasons why developing increasingly higher levels of thinking might not always be advantageous? To understand the reason for the current distribution of higher order thinking levels in humans, I want to shed light on the sensitivity of evolutionary dynamics of higher-order thinking agents, by analysing an agent-based model of higher-order thinking agents over many generations.

The remainder of this paper is structured as follows: Section 1 provides a literature review of influential papers that address similar or related research questions, summarizing their findings. Section 2 explores the concept of Theory of Mind (ToM) and its modeling in a game-theoretical context, forming the basis for understanding the design of the working model of this study, described in Section 3. Section 4 details the simulation employed and it´s results. Following up, Section 5 summarizes the key results and Section 6 interprets them.

1 Literature Review

The exploration of strategic sophistication and higher-order thinking within populations has been addressed in various studies, providing insights into why such cognitive abilities are not more widespread, despite their potential advantages.

Stahl's "Evolution of Smart-n Players" (1992)[27] provides a foundational understanding of the evolutionary dynamics behind strategic sophistication. Stahl argues that while advanced reasoning in games might offer a survival advantage, there are specific conditions under which simpler strategies can be equally effective. These include the presence of manifest optimal strategies, where the best course of action is obvious; the cognitive or resource costs associated with sophisticated strategies; and an upper limit to the benefits of additional cognitive sophistication. This framework aligns with the central question of my thesis, exploring why higher-order thinking remains rare despite its potential advantages. Stahl's work underscores the role of cognitive costs and diminishing returns, suggesting that both environmental constraints and strategic factors shape the distribution of thinking levels in populations.

Similarly, Lenaerts et al.'s "Evolution of a Theory of Mind" (2024)[17] investigates the evolution of cognitive traits like Theory of Mind (ToM), which involves understanding others' mental states. Their study reveals that bounded reasoning strategies often dominate in evolutionary contexts because they balance cognitive effort with strategic effectiveness. Moreover, the optimism bias, where individuals overestimate the likelihood of favorable outcomes, can foster cooperation and stabilize behavior in social interactions. This is particularly relevant to my research, as it suggests that higher-order thinking may not only be limited by cognitive effort but also by the evolutionary advantages of simpler, less cognitively demanding strategies. Their work illustrates how strategic trade-offs and cognitive limitations influence the evolution of reasoning abilities, offering valuable insights for my analysis of the evolutionary dynamics of higher-order thinking agents.

De Weerd, Verbrugge, and Verheij's study, "How Much Does It Help to Know What She Knows You Know?" (2013)[11], delves into recursive reasoning in competitive environments using agent-based models. Their findings suggest that while first- and second-order reasoning offer substantial benefits, the returns from moving to higher-order reasoning are diminishing. This insight is crucial for understanding the limited distribution of higher-order thinking in human populations. The cognitive and strategic trade-offs highlighted in this study emphasize that while recursive reasoning can be advantageous up to a certain point, its increasing complexity and associated costs might explain why it does not evolve universally.

Robalino and Robson's "The Evolution of Strategic Sophistication" (2016)[24] further explores the evolutionary foundations of strategic reasoning. Their work suggests that strategic sophistication—particularly the ability to infer and adapt to others' preferences—can offer significant advantages in novel strategic environments. However, these benefits come at a cognitive cost, suggesting a trade-off between strategic advantage and resource expenditure. This aligns with my research by highlighting that, while higher-order thinking may provide adaptive advantages, its cognitive demands limit its widespread adoption in populations.

Kimbrough, Robalino, and Robson's "The Evolution of Theory of Mind: Theory and Experiments" (2014)[15] extends the analysis of strategic cognition by investigating how the capacity to understand others' preferences (Theory of Mind) evolves in dynamic, strategic environments. Their findings suggest that ToM is a valuable adaptive trait, as agents capable of inferring others' preferences perform better in complex interactions. However, this ability is resource-intensive, which limits its evolution in the population. This paper supports my research by emphasizing that the evolution of sophisticated cognition, like ToM, is influenced by the trade-offs between the benefits of strategic adaptation and the costs of developing such cognitive abilities.

Mohlin's "Evolution of Theories of Mind" (2012)[19] also addresses the dynamics of reasoning levels in populations engaged in repeated games. He argues that higher-order reasoning often dominates in certain types of coordination games but coexists with simpler strategies in others, where strategic diversity can lead to beneficial equilibria. The introduction of partial observability—where agents can infer the reasoning levels of others—adds complexity and supports the coexistence of various cognitive types. Mohlin's findings suggest that the distribution of reasoning levels is not solely the result of cognitive limitations but also reflects strategic benefits, reinforcing the idea that the prevalence of higher-order thinking is shaped by both evolutionary pressures and the structure of the strategic environment.

These studies collectively provide valuable perspectives for understanding the evolution of higher-order thinking in human populations. They demonstrate that the distribution of cognitive sophistication is influenced not only by the cognitive and resource costs of advanced reasoning but also by the strategic advantages and limitations inherent in different game environments. By examining the balance between cognitive effort and evolutionary payoff, these works offer critical insights into the evolutionary dynamics that may explain why higher-order thinking remains limited in many human populations.

To sum it up: the current literature suggests that higher-order thinking and strategic sophistication in populations are limited by a combination of cognitive costs, diminishing returns, and evolutionary trade-offs. Advanced reasoning, while potentially advantageous, often incurs significant cognitive effort, which can outweigh its benefits in many scenarios. Studies highlight that simpler strategies frequently dominate due to their efficiency and adaptability in evolutionary contexts (Stahl, Lenaerts et al., Robalino & Robson[17, 27]). Additionally, recursive reasoning beyond first or second order offers diminishing returns, making higher levels less prevalent in populations (De Weerd et al.[11]). The trade-offs between strategic benefits and resource demands explain why sophisticated traits like Theory of Mind are limited in their evolutionary spread (Kimbrough et al., Mohlin[15]). While these studies provide diverse perspectives, they converge on the idea that the evolution of higher-order reasoning is shaped by balancing cognitive effort with strategic advantage.

Now that we have an overview of the current state of research regarding the question of how the distribution of ToM abilities among humans can be explained, we will delve into what Theory of Mind (ToM) is and how it can be modeled in game theory, in order to understand the working model of this paper.

Part II

Theoretical Background

2 Theory of Mind

In the realm of cognitive science and psychology, the Theory of Mind (ToM) stands as a foundational concept illuminating the intricacies of human social interaction and understanding. At its core, ToM encapsulates the capacity of individuals to attribute mental states—beliefs, intentions, desires, emotions—to oneself and others, thus enabling the interpretation and prediction of behavior based on these attributions.

Central to the Theory of Mind framework is the recognition that individuals do not passively perceive the world around them but actively engage in the interpretation and inference of others' mental states. This cognitive process allows individuals to anticipate and respond to the behavior of others, fostering empathy, cooperation, and effective communication within social groups.

The Theory of Mind concept has garnered significant attention and research across various disciplines, including psychology, neuroscience, philosophy, and artificial intelligence. For example, Halford (1993)[14] examines the role of Theory of Mind (ToM) in couples' behavioral therapy. Estes, Wellman, and Walley (1989)[12] explored how children think about mental phenomena, and Perner (1991)[23] addressed the question of whether autistic children lack a ToM. These are just a few illustrative papers showcasing the breadth and diversity of ToM research. This is unsurprising, as ToM serves as a cornerstone for understanding diverse phenomena such as social reasoning, empathy, deception, and cultural differences in social cognition—all of which are crucial factors influencing an individual's ability to thrive and reproduce within a population.

2.1 Theory of Mind and Evolutionary Game Theory

Unsurprisingly, Theory of Mind also plays a crucial role in the domain of (evolutionary) game theory, where strategic interaction and decision-making unfold within complex social contexts. By integrating insights from Theory of Mind into game-theoretic models, researchers aim to elucidate how individuals reason about the beliefs, intentions, and actions of others—a pursuit essential for unraveling the dynamics of strategic behavior in competitive and cooperative settings.

Theory of Mind contributes to understanding the dynamics of natural selection and the emergence of cooperative behaviors. In evolutionary settings, individuals with a more sophisticated Theory of Mind may be better equipped to assess the intentions and strategies of their peers, leading to differential reproductive success based on social acumen. Thus, Theory of Mind provides a nuanced lens through which to explore the evolution of social behaviors and their consequences for the survival and proliferation of individuals across generations.

Two Dimensions of Theory of Mind It is important to distinguish our research question "Why did our higher level thinking develop as it did?" from other research questions about the evolutionary background of ToM: Brüne and Brüne-Cohrs (2006)[6] suggest that Theory of Mind (ToM) evolved in response to social selection pressures in the environment of our primate ancestors and is now highly developed in modern humans. They focus on the ability to recognize and accurately interpret the emotions and intentions of other members of one's species. In my paper, however, I will focus on a different aspect of Theory of Mind: rather than examining the precision of the individual thinking step, In my simulations, I assume that all agents make correct reasoning steps, differing only in the depth of their higher-order thinking. My focus is on the usefulness of Theory of Mind (ToM), particularly higher-order thinking, in competitive scenarios.

To make this more clear: There are two dimensions to assess the higher-order thinking capabilities of an agent. The Dimension Brüne and Brüne-Cohrs (2006)[6] focus on is the precision of a single thinking step. For example, if I reason what an agent in front of me is thinking my interpretation of their body language can be more or less accurate. This accuracy is also a measure for ToM. But this part of ToM is hard to explore with game theoretical tools and also hard to integrate into an agent- based model, as the one this paper is based on. Therefore this paper focuses on the second Dimension of ToM: How many steps an agent can perform in a recursive thinking task, as already illustrated in the goalkeeper exampleI.

Shift from superintelligence to bounded rationality For a long time, it was not possible to model Theory of Mind (ToM) in game theory because it was standard to assume that agents had common knowledge of rationality. This meant they were capable of performing infinitely many recursive thinking tasks, leading them to the Nash equilibrium solution, if one existed. In the following, we will review the methodology of the most important papers that moved game theory away from the standard Nash equilibrium approach and made it possible to model different levels of cognitive sophistication.

One of the most influential papers, building on Bernheim and Pearce (1984)[3], is Stahl (1993)[28]. Stahl examines how agents with different levels of sophistication perform in an evolutionary game theory context. The question he addresses is very similar to the one I focus on in my work: Do more intelligent agents outperform less intelligent ones? The setup of his experiment is as follows: He lets different participants compete against each other in simple two-player matrix games. The goal is to observe whether they choose their strategies according to the k-rationality model or the Nash equilibrium and to determine whether players with higher k-levels outperform those with lower k-levels.

Stahl's model of k-level agents differs in one key aspect from Bernheim and Pearce (1984)[3]: All k-level agents are fully informed about the distribution of higher-order thinking levels in the population. Therefore, instead of simply responding optimally to a k-1 agent, they use this information about the distribution to determine their actions.

Bernheim and Pearce (1984)[3] and Stahl (1993)[28] agree on many aspects of their respective models, which both aim to model Theory of Mind (ToM). Both agree that rationality does not require an agent to choose the Nash equilibrium. According to Bernheim (1984)[3], Nash equilibrium does not solve the problem of non-cooperative strategic choice. Nash behavior is neither a necessary outcome of rationality nor a reasonable empirical assumption.

The main reason for this is that agents in real life are not super-intelligent, meaning there is no "common knowledge of rationality," as assumed in much of game theory research up to that point. This was made especially clear by Nagel's field study (1995)[21], which showed that agents do not act as if common knowledge of rationality exists in real competitive scenarios. Nagel (2019)[21] even goes so far as to suggest that the classic Rational Expectations Equilibrium, which assumes that all agents have perfect knowledge of their environment and make, on average, correct decisions (Muth 1961)[20], is not sufficient as a foundation for macroeconomics anymore because too many empirical studies have challenged it.

This shift in game theory led to the development of models that allow agents to be modeled without assuming they think infinitely about their opponents' strategies or assume their opponents are super-intelligent. As a result, various levels of Theory of Mind can also be modeled in game theory.

2.2 Popular Concepts to Model ToM in Game Theory

2.2.1 K-rationalizability

In order to understand K-Rationalizability we first have to take a look at the Rationalizability concept in Game theory. Rationalizability is a solution concept proposed by Bernheim and Pearce (1984)[3] that requires common knwoledge of rationality. Meaning every player has to be rational and has to know that every other player is rational and has to know that every other player knows that every other player is rational and so on ad infinitum. A strategy is rationalizable if there exists at least one set of believes of the players about each other actions that would lead to this strategy being played. Meaning a strategy is rationalizable if there exists a set of believes which makes it a best response for both players. Note that the assumption of common knowledge of rationality restricts the believes which players can have. For example could a player not believe that his opponent would play a strategy that will never be a best response any action the player can choose. To determine for which strategy-combination such a set of believes exists, we use the iterated elimination of dominated strategies, as illustrated in Figure 1.

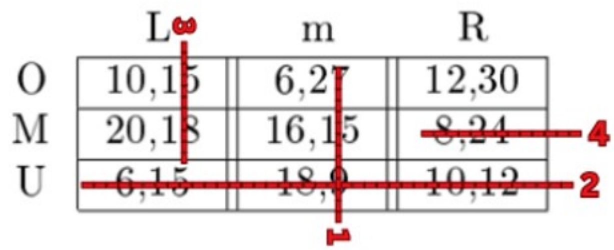


Figure 1: Iterated elimination of dominated strategies (IESDS)

Remove all dominated actions from the game matrix, because there is no believe for which it makes sense to play those strategies, because they can by definition never be a best response. Now you have a new smaller game matrix, also remove all dominated strategies from this matrix. Repeat this procedure untill there is no dominated strategy in the matrix anymore. The strategys that are not eliminated by this procedure are called rationalizable.

K-Rationalizability is closely related to the Rationalizability concept: while Rationalizability is fulfilled by all actions that survive infinite steps of iterated elimination of dominated strategies, k-rationalizability is fullfilled by all actions that survive finite (k) steps of iterated elimination of dominated strategies. Therefore k-rationalizability does not rely on common knowledge of rationality. It relies on the knowledge of rationality but not ad infintum, only to a certain depth.

Unlike strict k-level reasoning (as in level-k models), where agents often hold fixed beliefs that opponents reason exactly at a specific level (e.g., Level-1 agents assume all others are Level-0), k-rationalizability allows for heterogeneity in beliefs (Battigalli & Siniscalchi, 2002)[1]. A k agent may consider the possibility of opponents reasoning at levels lower than k−1, introducing tolerance for varying degrees of sophistication.

Thus while a k-agent in a k-rationalizability model with the cognitive ability to perform k iterations of IESDS is required to play only within the k-rationalizable set, the agents can adapt to different opponent levels by selecting different mixed strategies within this set when it is strategically advantageous[1].

Suppose an agent engages in a two-player game and, after performing two iterations of IESDS (k=2),

[1] If the model would not grant the agent the ability to strategically select his actions out of his k-rationalizable strategies, the heterogeneity in beliefs would not change the action of the player at all and would therefore be superfluous.

identifies two remaining k-rationalizable strategies, S1 and S2. The agent believes their opponent is a naive player who might not fully adhere to the principles of rationality, possibly playing a dominated strategy S3, for which the best response is S3, which is outside the k-rationalizable set. In this case, while the agent cannot directly choose S3, they may assign probabilities in a mixed strategy to S1 and S2 that maximize their utility, anticipating the opponent's mistake. For instance, the agent might favor S1 heavily if S1 exploits the opponent's expected deviation more effectively than S2. This flexibility within the k-rationalizable framework allows agents to strategically adapt while still adhering to the theoretical bounds of rationalizability.

However, due to this dynamic flexibility, it is possible that fixed-point reasoning could come into play, but only to a limited extent: Even though in (two-person games), a player can determine his set of k-rationalizable strategies via k rounds of iterated strict dominance, without the need for fixed-point reasoning (Crawford et al 2013[10]), his strategy selection within the k-rationalizable set might still be dictated by fixed point reasoning in multishot games: Because the agent's strategy selection can vary depending on their received information about the opponent, their belief system may need to stabilize in a way that approaches a fixed point, albeit less rigorously or completely than in models that rely on strict fixed-point reasoning (e.g. cognitive hierarchy models).

Due to these more realistic assumptions[2], k-rationalizability has a wide range of applications in various fields: For instance, in economics, k-rationalizability has been employed to analyze bidding behavior in auctions with private information (Battigalli & Siniscalchi, 2003[2]). In political science, it has shed light on the strategic interactions among political actors with differing levels of rationality (Feddersen & Pesendorfer, 1998[13]). Additionally, in computer science, k-rationalizability has been utilized to design algorithms for multi-agent systems with bounded rationality (Wooldridge, 2009[32]). In agent-based simulations, incorporating k-rationalizability enhances realism by capturing the diverse levels of rationality present in real-world decision-making. This approach allows for a broad spectrum of behaviors and strategies among agents, reflecting the heterogeneous nature of human decision-making.

2.2.2 k-level reasoning

K-level reasoning is a theoretical framework in game theory closely associated with the Theory of Mind. It offers an economical approach to integrating Theory of Mind into game theoretical models by attributing agents' beliefs about the beliefs of other agents. For example: Player 1 anticipates player 2's potential action x and responds optimally with an action b(x). To outsmart player 1, player 2 must accurately predict player 1's beliefs leading to b(x) and counter with $b^2(b(x))$... This notion of hierarchical thinking was initially

[2] more realistic then the assumptions of rationalizability. (common knowledge of rationality)

introduced by Nagel (1995)[21]. Level-k reasoning, often denoted as k-order thinking, elucidates the depth of recursive strategic thinking in decision-making. The "k" in level-k signifies a specific level of strategic depth. Each level corresponds to a distinct mental model individuals employ to forecast others' actions and make decisions in interactive settings. Level-0 thinking denotes a non-strategical approach where individuals solely consider their immediate gains, disregarding others' cognitive processes. Level-1 thinking involves accounting for opponents' decision-making, assuming they are level-0 thinkers. Level-2 thinking extends this process, presuming opponents are level-1 thinkers. Individuals employing level-2 thinking contemplate how opponents would reason about their decisions, assuming opponents reason at level-1. This recursive logic persists for higher levels. So all in all a level k-agent best responds to the action of a level k-1 agent. Thus k-level reasoning does not require fixed-point reasoning because the beliefs about the opponent's strategies are deterministic, and therefore there is no need and no possibility for beliefs to converge to a fixed point: In k-level reasoning, agents are not able to iteratively update their beliefs or strategize toward a fixed point. This makes it less cognitively taxing compared to for example cognitive hierarchy models, which rely on fixed-point reasoning in multi-shot games. Due to this inability of the agent to adapt to the opponent, the realism of k-level reasoning is limited in multishot games. Since agents do not update their beliefs about the opponent's level or strategy based on past actions, they can not correct wrong believes about the opponents level or action and cannot account for dynamic changes in the opponents strategy.

Additionally the static believes lead to k-level reasoning also being k-rationalizable: Level 0 are assumed to randomize, and agents at Level 1's best response will therefore never be a dominated strategy because otherwise it would not be a best response by definition (Camerer et al., 2004[8]). Level 2 agents, when assuming they play exclusively against Level 1 players who have eliminated dominated strategies, will not play strategies that are dominated after the dominated actions for the Level 1 players are removed from the game. This logic can be extended to iteratively eliminate as many dominated strategies as desired if k-step thinkers assume that their opponents are $k - 1$. Essentially, a k-level agent's strategy choice can be seen as a subset of the strategies that would be considered rationalizable in a k-rationalizability framework. K-level reasoning can be viewed as a special case or a refinement of k-rationalizability, where the agent's deterministic beliefs about the opponent ensure that their strategy is on the one hand static and on the other hand consistent with the rationalizable set of strategies.

2.2.3 Cognitive Hierarchy models

The Cognitive Hierarchy (CH) model offers a framework for understanding decision-making dynamics in social interactions by acknowledging the diverse cognitive capacities individuals possess. At its core, this model proposes that within a population, individuals exhibit varying levels of cognitive sophistication, influencing their strategic decision-making processes.

One fundamental aspect of the CH model involves stratifying individuals into different cognitive levels, often represented as Level 0 (L0), Level 1 (L1), Level 2 (L2), and so forth. These levels delineate degrees of cognitive prowess, with higher levels presumed to possess greater analytical depth and strategic foresight. Consequently, individuals at higher levels are expected to make more accurate predictions about the behavior of others, leveraging their enhanced cognitive abilities to assess the strategic implications of their decisions.

However, this heightened cognitive acumen comes at a cost. Higher-level players are assumed to incur greater cognitive effort or "thinking costs" when deliberating on their strategies, reflecting the mental exertion associated with deeper reasoning. Thus, while higher-level players may excel in making nuanced strategic choices, they also bear the burden of increased cognitive demands.

A key tenet of the CH model lies in its consideration of players' beliefs regarding the distribution of cognitive types within the population. Each player believes they understand the game better than the others, and a player with reasoning ability k believes the reasoning abilities of other agents are distributed according to a certain distribution function, often assumed to be a Poisson distribution (Camerer et al., 2004[8]): Step k thinkers assume that their opponents are distributed according to a normalized Poisson distribution, from step 0 to step k - 1. This means they accurately predict the relative frequencies of players performing fewer steps of thinking but ignore the possibility that some players may be performing as much or more.

In contrast to k-level reasoning, Cognitive Hierarchy (CH) models do not have deterministic beliefs, which necessitates the use of fixed-point reasoning. In multi-shot games, agents in CH models need to iteratively update their beliefs to ensure their expectations about the opponent's level of reasoning align with the opponent's observed actions. In CH models, the level-1 agent's belief is updated based on the assumption that the opponent's level is lower, which is typically done through Bayesian updating (Crawford et al 2013[10]). This process continues until the agent's expectations about the opponent's behavior converge with the observed actions.

In CH models, a Lk agent operates similarly to a level-k agent modeled with k-level reasoning. However, as the level increases, a CH agent might diverge from its corresponding k-level agent counterpart. While both CH L1 agents and level-k level-1 agents make undominated choices, a CH agent at level-k may not necessarily follow the principles of k-rationalizability, unlike a level-k agent (Crawford et al 2013[10]). This is because the belief distributions in CH models opens up the possibility that a Lk Agent believes to play against an agent Lk-x with $x \in [2, k]$. Thus enables him to play an optimal response that is not in the set of k-rationalizable actions but in the broader set of $k-(x+1)$-rationalizable actions. Therefore an action of an Lk agent must not be k-rationalizable.

There is, however, one exception when Lk agents can be assumed to play k-rationalizable strategies: When the agent's level is significantly lower than the parameter of the Poisson distribution. In this case, the agent will most likely think that their opponent is one reasoning level below them (Camerer et al., 2004[8]). A similar explanation shows why, according to a cognitive hierarchy model, higher reasoning levels do not always bring more benefits compared to lower ones. If all players believe the levels of others follow a Poisson distribution with a sufficiently large lambda, with levels lower than their own, as k increases, the beliefs of a Lk and Lk-1 player will be the same. Since the actions in this model depend entirely on the beliefs about the other player, Lk and Lk-1 players will take the same action when k is large enough. Therefore, at this limit, a higher k is not beneficial (Camerer et al., 2004[8]). However, since empirical data shows that most people's higher-order thinking levels are lower than 4, this exception is less relevant in reality.

Part III

Methods

3 The Working Model

After reviewing the various ways to model Theory of Mind in game theory, we now turn to the agent-based model (ABM) used in the simulation central to this paper. An ABM is particularly appropriate because it enables the detailed representation of individual agents with different reasoning levels and decision-making processes within a dynamic, multi-agent environment. ABMs are especially effective at capturing interactions among heterogeneous agents, making them ideal for studying the evolution and impact of reasoning levels in dominance-solvable games.

In the simulation, agents play dominance-solvable games where, through four iterations of the iterative elimination of strictly dominated strategies (IESDS), the game matrix is reduced to its unique Nash equilibrium. Level-0 agents adopt a naïve approach, randomizing equally over all actions. This aligns with the modeling approach suggested by Camerer et al. (2013)[7] for cognitive hierarchy (CH) models, which use Level-0 strategies to anchor the beliefs of Level-1 and higher-level agents. Level-1 agents are less naïve, as they exclude strictly dominated strategies from their mixed strategy. Their remaining actions are distributed uniformly among strategies that are not strictly dominated in the original matrix.

Unlike CH models, the agents in this simulation have no heterogeneity in beliefs; Level-k agents always assume that they are playing against Level-k−1 agents. This epistemic structure, inspired by k-level reasoning models, ensures that agents' beliefs are static and thus avoids the computational challenges of fixed-point reasoning. This simplification is especially beneficial due to the large amount of multi-shot games in my simulation, where avoiding complex iterative belief updates increases simulation efficiency.

Although this approach is less realistic, it aligns with Wooldridge's (2009)[32] emphasis on balancing theoretical soundness with computational feasibility in agent-based modeling. By restricting agents' decision-making to k-rationalizable sets, the model focuses on strategic feasibility without incurring high computational costs.

Level-k agents can perform the kth iteration of IESDS by considering which strategies Level-k-1 agents have eliminated. So level-k agents uniformly randomize among k-rationalizable strategies, reflecting a form of k-rationalizability modeling. Thus, they cannot optimize their expected utility by selecting specific mixed strategies within the k-rationalizable set when it might be strategically advantageous. This model's design balances computational efficiency with the theoretical principlesof cognitive hierachy models, k-rationalizability and k-level reasoning.

4 Simulation

In the first generation, 200 agents are assigned a respective level drawn from either a manually-set or a Poisson distribution[3] (see Source Code 6), with the parameter lambda being a selectable variable, so that each agent is assigned a level between 0 and 4.[4] The number of agents and the number of games in the environment can be specified through the simulation's variables (see Source Code 1). In each generation, every agent plays each game against all other agents.

All Games are based on one gamestructure which is uniquely solvable with iterated elimination of dominated strategies. You see an example for a game with this structure in Table 1:

		Player 2		
		L	m	R
Player 1	O	x1,c1	x2,a1	x3,b1
	M	y1,c2	y2,a2	y3,b2
	U	z1,c3	z2,a3	z3,b3

		Player 2		
		L	m	R
Player 1	O	10,15	6,27	12,30
	M	20,18	16,15	8,24
	U	6,15	18,9	10,12

Table 1: First Iteration

In the following I will lay out the conditions for the utility values $a_i, b_i, c_i, x_i, y_i, z_i$ of the basic game structure. Based on this structure the simulation will randomly generate games with different utility values. The reason why not only chose one game with this structure is to avoid that random utility values of that one game influencing the result of the simulation.

4.1 Determining the basic game structure

To determine the structure I will go through each iteration of the IESDS process, illustrated in one specific game of that structure for then to elaborate which conditions must be met for the general game so that the dominated elimination process can take place. Finally I will explain which implications this elimination is having on the actions of the k-level agents playing such a game.

[3]According to Camerer et al. (2004), the distribution of higher-order thinking levels is best described by a Poisson distribution

[4]It is possible to draw values greater than 4 when using a Poisson distribution. In this simulation, agents with levels above 4 are categorized as Level 4 agents. This is not problematic, as agents with levels higher than 3 behave the same way in the focus games of this simulation, meaning they select the same strategy. Therefore, Level 4 agents can be seen as Level k agents for k in the range of [4, ∞].

		Player 2		
		L	m	R
Player 1	O	x1,c1	x2,a1	x3,b1
	M	y1,c2	y2,a2	y3,b2
	U	z1,c3	z2,a3	z3,b3

		Player 2		
		L	m	R
Player 1	O	10,15	6,27	12,30
	M	20,18	16,15	8,24
	U	6,15	18,9	10,12

Table 2: First Iteration

First Iteration The first iteration eliminates strategy m for Player 2[5] because it is strictly dominated by strategy R. In other words, for every action Player 1 takes, the utility of playing m for Player 2 is always less than the utility of playing R. This means that, in the general game, for all actions of Player 1, $a_i < b_i$ must hold true. Additionally, strategy L is not dominated by R, and vice versa. Therefore, there must exist at least one instance where c_i is greater than b_i, and at least one instance where bi is greater than ci. Implications for agents: All Level 1 agents in the role of Player 2 will not play strategy m, because, by definition, they are able to identify and avoid dominated strategies. Instead, they will randomize uniformly between strategies L and R. On the other hand, Level 1 agents in the role of Player 1 will not adjust their strategy due to this first elimination. This is because no strategy is dominated for them. As a result, they are indifferent between all possible actions and will randomize uniformly between them, just like a Level 0 Player 1. The fact that no strategy is eliminated for Player 1 is guaranteed by the following conditions:

1. $y_1 > x_1$ and $y_1 > z_1$: This condition ensures that M cannot be dominated.

2. $z_2 > x_2$ and $z_2 > y_2$: This condition ensures that U is not dominated.

3. $x_3 > y_3$ and $x_3 > z_3$: This condition ensures that O is not dominated.

Of course there are other possibilities to make sure that in all games of this structure, no strategy for Player 1 is dominated in the first round of elimination, this is just the one I chose.

		Player 2		
		L	m	R
Player 1	O	x1,c1	x2,a1	x3,b1
	M	y1,c2	y2,a2	y3,b2
	U	z1,c3	z2,a3	z3,b3

		Player 2		
		L	m	R
Player 1	O	10,15	6,27	12,30
	M	20,18	16,15	8,24
	U	6,15	18,9	10,12

Table 3: Second Iteration

Second Iteration The second iteration is eliminating U for the Rowplayer, because after the first elimination action m is not longer part of the game matrix and now U is strictly dominated by O. Therefore $x_1 > z_1$

[5] "Player 2" is used interchangeable to "Columnplayer" and "Player 1" to "Rowplayer"

and $x_3 > z_3$ must be the case in the general game. Also, it is not the case that O and M are dominated by each other therefore it must be the case that $y_1 > x_1$ and $x_3 > y_3$ or that $y_1 < x_1$ and $x_3 < y_3$. We also want for Player 2, that the actions L and R do not dominate each other, but we already determined this in the first elimination round.

Implications for actions of agents: Agents of level 2 that are in the role of Player 1 assume their opponent (Player 2) to be level 1, therefore they know he will randomize between L and R and they now see that U for them is dominated. Thus Agents in the role of Player 1 will randomize between M and O. Agents of Level 2 that are in the shoes of Player 2 still play a mix between L and R because they assume Player 1 to be level 1 and a Player 1 will randomize between all actions as we found out.

		Player 2		
		L	m	R
Player 1	O	x1,c1	x2,a1	x3,b1
	M	y1,c2	y2,a2	y3,b2
	U	z1,c3	z2,a3	z3,b3

		Player 2		
		L	m	R
Player 1	O	10,15	6,27	12,30
	M	20,18	16,15	8,24
	U	6,15	18,9	10,12

Table 4: Third Iteration

Third Iteration The third iteration is elimination L for the Columnplayer, because after the second elimination Player 1 can not choose M anymore and now R is dominating L. That is $b_1 > c_1$ and $b_2 > c_2$.

Implications for actions of agents: Agents level 3 in role of Player 2 assume their opponent to be level 2 which implies he is mixing between O and M. When Player 2 knows his opponent will only mix between O and M, R is the dominating strategy for him. Agents with level 3 in role of Player 1 will think their opponent as an level 2 Columnplayer is playing a mix between L and R and therefore still play a mix between O and M.

		Player 2		
		L	m	R
Player 1	O	x1,c1	x2,a1	x3,b1
	M	y1,c2	y2,a2	y3,b2
	U	z1,c3	z2,a3	z3,b3

		Player 2		
		L	m	R
Player 1	O	10,15	6,27	12,30
	M	20,18	16,15	8,24
	U	6,15	18,9	10,12

Table 5: Fourth Iteration

Fourth Iteration The fourth iteration eliminates Strategy M for Player 1, therefore $x_3 > y_3$ must be the case.

Implication for the agents: Agents in role of Player 1 with level 4 expect their opponent to be level 3 and therefore, as elaborated above, to play R. Therefore they will choose O as their strategy. Agents level 4 in the shoes of the Columnplayer will assume their opponent to play a mix between O and M and therefore play R which is the best response because as above R dominates in this constellation.

After the fourth iteration the iterated elimination of dominated is over, simply because there is only one strategy for each player left, therefore no strategy can be dominated by another strategy. Now we take a look at the conditions of the structure, we gained:

$$\forall i : a_i < b_i \tag{1}$$

$$\exists i : c_i < b_i \tag{2}$$

$$y_1 > x_1 \wedge y_1 > z_1 \tag{3}$$

$$x_3 > y_3 \wedge x_3 > z_3 \tag{4}$$

$$z_2 > y_2 \wedge z_2 > x_2 \tag{5}$$

$$z_1 < x_1 \wedge z_3 < x_3 \tag{6}$$

$$y_1 > x_1 \vee y_3 > x_3 \tag{7}$$

$$c_1 < b_1 \wedge c_2 < b_2 \tag{8}$$

$$x_3 > y_3 \tag{9}$$

By using logical transformation laws (see proof 6 in Appendix) we can deduct following main conditions, which define the structure of the game:

$$c_1 < b_1 \wedge a_1 < b_1$$

$$c_2 < b_2 \wedge a_2 < b2$$

$$a_3 < b_3 < c_3$$

$$z_1 < x_1 < y_1$$

$$z_2 > y_2 \wedge z_2 > x_2$$

$$x_3 > y_3 \wedge x_3 > z_3$$

Based on these conditions the simulation randomly generates a choosable amount of games (see Source Code 4).

After every agent played every game against every other agents the utility of the agents is fed into following Replicator dynamics (see Source Code 2):

4.2 Replicator dynamics

The utility of an agent i is equal to the sum of all utilities obtained in games against all other agents in the population:

$$\mathrm{U}_i = \sum_{j \neq i} \mathrm{U}_{i,j}$$

The sum of all agents i possessing the property of being at level k is summed up to the k-level utility:

$$\sum_{i \in K} \mathrm{U}_i = \mathrm{U}_k$$

The distribution of k-agents in the next generation consists of the proportion of the respective k-level utility to the utility of all U_k, where k is an element of $[0, 4]$:

$$p^k = \frac{\mathrm{U}_k}{\sum_{k=0}^{4} \mathrm{U}_k}$$

The Number k-agents in the next generation is calculated by multiplying the distribution times the total number of individuals:

$$N_{k \in K}^{g+1} = p^k * \sum_{k=0}^{4} N_{k \in K}^{g}$$

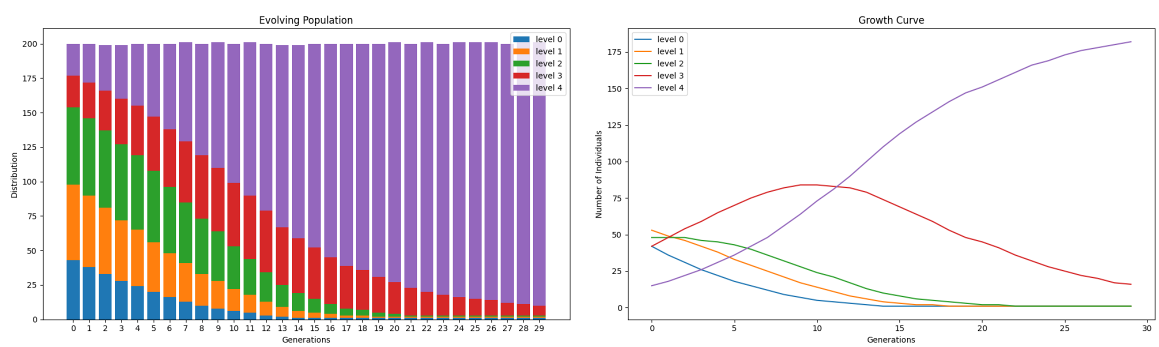
After the replicator dynamics determined the next generation, the simulation process as described starts all over again. This goes on and on until as many generations passed as determined in the variables of the code. In the end, the code outputs a column chart that illustrates how the distribution of the individuals in the population changed over time and a graph with functions illustrating the growth of the number of each agent type (see Appendix 7).

Part IV

Results

In the following, I present the results of the simulation with different parameters. The parameters that were varied between the simulations are the lambda (λ) parameter of the Poisson distribution, the cost function for higher-order thinking depending on the thinking level k, and the binary setting determining whether Level 0 agents have the ability to maximize their expected utility. The Y-axis shows the number of individuals, which is set to 200 in total[6]. The X-axis indicates the number of each generation, starting with the 0th generation up to the 29th generation. Each column represents the population distribution in the respective generations, where each color indicates the proportion of agents with a specific thinking level in the population, as shown in the legend. In the second graph, the number of level k agents is shown as a function depending on the generation. This makes it easier to observe the growth of the different agent types.

4.3 Simulation A



(a) (b)

Figure 2: Poissondistribution (λ=1.61); c(k)=0; Level 0 not clever

The first simulation, shown in Figure 2, depicts a population that initially follows a Poisson distribution with $\lambda = 1.61$. In this simulation, there are no costs associated with higher thinking levels, and Level 0 agents select random actions instead of choosing the action with the highest expected value.

After 30 generations, it becomes evident that Level 4 agents have the best conditions for reproduction. Only Level 3 and Level 4 agents remain in the population by the final generation[7]. As shown in further

[6] Due to rounding, the population size in some generations is 199 or 201, which does not significantly affect the overall result and also resembles a real population that may fluctuate slightly around its carrying capacity

[7] Due to rounding, each type of agent, even after extinction, leaves behind a single individual in the population. This appears

simulations, Level 3 agents eventually die out entirely when coexisting with Level 4 agents.

Although Level 4 agents are the long-term winners, the rate at which other agents go extinct depends strongly on their level. The number of Level 3 agents increases during the early generations (up to around Generation 9), though not as quickly as the proportion of Level 4 agents. For comparison: in Generation 0, there are about 20 Level 4 agents, which grows to around 70 by Generation 9. The proportion of Level 3 agents starts at approximately 35, also reaching about 70 in Generation 9, at which point the Level 4 agents overtake it.

The proportion of Level 2 agents appears stable during the first three generations but then slowly declines until they are nearly completely displaced by Generation 18. This occurs long before the extinction of Level 3 agents (after Generation 30) but after Levels 0 and 1 have died out. Notably, Levels 0, 1, and 2 agents go extinct in quick succession, within roughly four generations (Generations 14–18).

Generally, the lower the agent level, the faster their population decreases from generation to generation. The time until extinction is negatively correlated with the thinking level. For example, even though there are more Level 0 and 1 agents than Level 2 agents in Generation 0, the Level 2 agents still outlast Levels 0 and 1. If we let N_k^g denote the number of k level individuals in the generation g, then we can formalize these two hypotheses mathematically:

Hypothesis I: $N_{k-x}^{g+1} - N_{k-x}^{g} < N_k^{g+1} - N_k^{g} \; for \, x \in [1, k]$

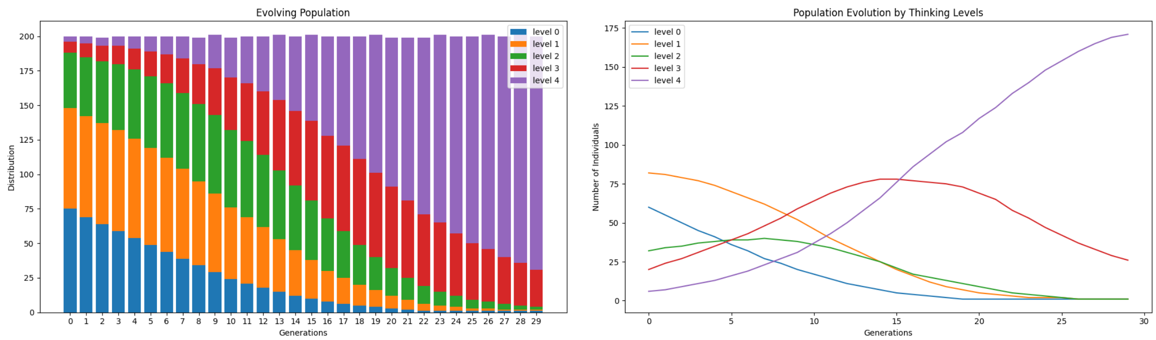
Hypothesis II: $N_k^{g'} = 0 \wedge N_{k-x}^{g''} = 0 \Rightarrow g' > g'' \; for \, x \in [1, k]$

Another notable observation in Figure 2b is that the curve for Level 4 agents intersects the peak of the curve for Level 3 agents. This means that in this simulation, the number of Level 3 agents begins to decline when there are more Level 4 agents than Level 3 agents.

All curves in Figure 2b exhibit a sigmoid shape, indicating logistic growth. However, this shape is very shallow for Level 0, 1, and 2 agents.

as a small line in the diagram but should not be given much attention.

4.4 Simulation B



(a) (b)

Figure 3: Poissondistribution (λ=1.0); c(k)=0; Level 0 not clever

The second simulation is shown in Figure 3. In this simulation, the first generation follows a Poisson distribution with lambda equal to 1. Similar to the first simulation, there are no thinking costs, and level 0 agents are not expected utility maximizers. The only difference is the lower lambda value (reduced by 0.5) in the Poisson distribution. Due to the lower lambda, there are more agents with lower levels and correspondingly fewer agents at levels 3 and 4 compared to the first generation. The number of level 2 agents is similar, but there are significantly more agents at levels 0 and 1. It is now interesting to observe how this change in the Poisson parameter affects population dynamics. Level 4 agents once again dominate the evolutionary competition, and it seems they will eventually take over the entire population (let N denote all individuals in a Population).

Hypothesis III: $\lim_{g\to\infty} N^g_{max(k)} = N$

However, the main difference compared to Simulation A is that it takes much longer for them to achieve this dominance. This delay could be due to the smaller initial number of level 4 agents in the first generation.

Hypotheses II and III also appear to be confirmed in this simulation: the lower the agent level, the faster the different agent types become extinct. Similarly, the extinction sequence starts with level 0 agents and ends with level 3 agents, even though there are far more agents with lower levels. One notable observation is that in the first generation of Simulation B, the number of level 2 agents is lower, yet they perform[8] better. This can be seen in their Overall Individual Share (OIS), which is 10.6% in Simulation A and 19.59% in Simulation B.

[8]"performance" is used equivalent to reproductive ability

Overall Individual Share The Overall Individual Share (OIS) is a metric developed specifically for this paper to measure the reproductive ability of level k agents. It calculates the proportion of level k agents out of all individuals that have existed across all generations [a]. For example, for level 4 agents, the OIS corresponds to the total area of the purple sections in all the bars of the bar chart. The OIS has an advantage over the Average Growth per 10 Generations (AG10), a metric that will be used later. The AG10 can be distorted by the early extinction of an agent type: it cannot differentiate between cases where an agent type goes extinct in the 5th generation or the 20th generation. In both situations, AG10 would calculate -100%×10÷G[b]. Furthermore, AG10 cannot distinguish between two scenarios: one where an agent type initially increases, then decreases sharply, and finally reaches a proportion x in the last generation, and another where it steadily declines to the same proportion x. In contrast, the OIS accounts for both the timing of extinction and temporary growth during the simulation.

[a] in evolutionary game theory, each generation is typically assumed to consist of entirely new individuals
[b] G stands for the number of Generations in the Simulation

This observation suggests that level 2 agents perform better in this simulation because there are fewer agents with higher levels (2+x) and more agents with lower levels (2−x)[9]. This leads to a general hypothesis: the number of k−x agents with x∈[1,k] positively affects the survival and reproduction of level k agents, while the number of k+x agents with x∈[1,∞) negatively affects them. Let $N_k{'}$ and $N_k{''}$ denote the number of level k agents in two different Populations:

Hypothesis IV : $N'^{\mathbf{g}}_{k-x} > N''^{\mathbf{g}}_{k-x} \wedge N_k{'} = N_k{''} \Rightarrow N'^{\mathbf{g}+1}_{k} > N''^{\mathbf{g}+1}_{k}$

Hypothesis V: $N'^{\mathbf{g}}_{k+x} > N''^{\mathbf{g}}_{k+x} \wedge N_k{'} = N_k{''} \Rightarrow N'^{\mathbf{g}+1}_{k} < N''^{\mathbf{g}+1}_{k}$

[9] From now on, agents with lower levels will be referred to as k-x agents, and those with higher levels as k+x agents.

Doesn´t Hypothesis IV entail Hypothesis V, vice versa?

If we consider the obvious hypothesis that the reproductive success of level k agents is positively influenced by the number of level k agents, then Hypotheses IV and V, along with this hypothesis, seem multicollinear. If an increase in k−x agents and k agents both enhance the reproduction of level k agents, it logically follows that reducing k+x agents would also increase the reproduction of k agents. This is because reducing k+x agents in a population of constant size would lead to an increase in the number of k and k−x agents. However, Hypothesis IV does not only account for the indirect effect of k+x agents (where fewer k+x agents indirectly lead to higher k-agent growth by increasing the k−x population). It also includes the direct effect that the number of k+x agents has on the growth of level k agents. Therefore, both Hypotheses IV and V remain valid even in a population that is not constant—for example, a population that has not yet reached its population limit.

Doesn´t Hypothesis I entail IV and V?

If a k-x_1 agent reproduces faster as an k-x_2 agent, which reproduces faster than an k-x_3 agent with $x_1 < x_2 < x_3$ this means, if we take a look at the replicator dynamics, the k-x_1 agent is getting the biggest share of the overall utility, the k-x_2 agent is getting the second biggest share and the k-x_3 the smallest share. Wouldn´t replacing the k-$x_1$with an k-x_3 agent automatically guarante an higher utility share and therefore an increased reproducitve ability for the k-x_2 agent? While it is true that the utility share for all agents together increase, when replacing an agent with a less reproductive agent, this does not necessarily means that a specifc type of agent also increases in their reproduction. When only taking Hypothesis I into accout it would be also plausible that the utility share caused by exchanging an k$-x_3$ for a k$-x_1$ agent, distributes only on k-x_4 agents and the k-x_2 reproduction stays the same. But when we also take Hypothesis IV and V into account we can conclude that exchanging an k$-x_3$ for a k$-x_1$ agent will increaise the relative utility of an k-x_2 agent and therefore his reproduction.

Another similarity with Simulation A is that the curve of level 4 agents once again intersects the peak of the curve of level 3 agents in Figure 2b. This suggests that the stagnation in the performance of level 3 agents is caused by the increasing number of level 4 agents. This observation supports Hypothesis V, as it shows that an increase in k+x agents reduces the reproductive success of level k agents.

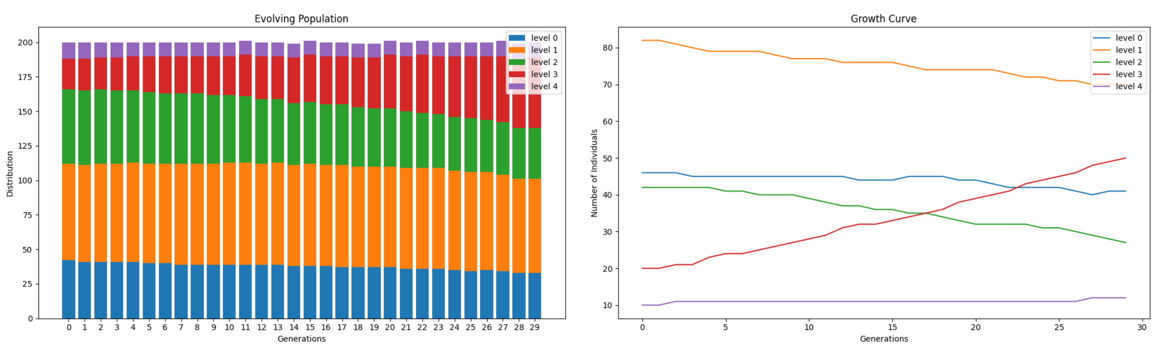
The growth functions also have a sigmoidal shape as in Simulation A, although the Level 0 growth function only shows very weak sigmoidal characteristics.

This raises the question: does the impact of k−x and k+x agents on the reproductive success of level k agents depend on the specific value of x? Or does the reproductive success of level k agents benefit equally from all k$-$x agents in the population and suffer equally from all k+x agents? To explore this, I will compare how the presence of different types of agents affects one another in Simulation D.

Question I: How does the number of different k$-x_i$ agents with $x_i \in$(1,k) influence the reproduction of k-agents?

This sigmoidal shape of the growth curves reflects the typical logistic growth observed within a population where no evolutionary equilibrium is established: Level 4 agents (and initially also Level 3 agents), as the dominant species, can grow exponentially until they reach the natural population limit. Meanwhile, the dominated species experience negative logistic growth as a result.

4.5 Simulation C and D



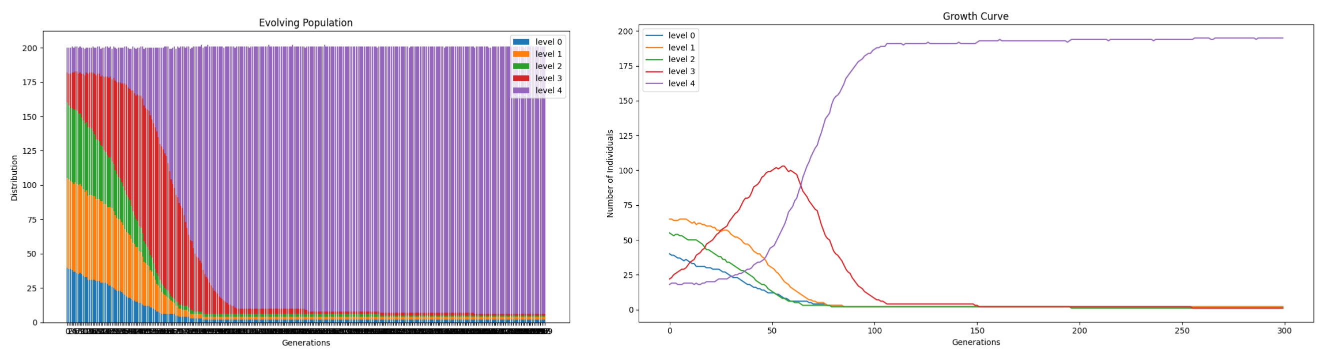
(a) (b)

Figure 4: Poissondistribution (λ=1.61); $c(k) = k^{1.2}$; Level 0 not clever

Figure 4 shows the results of a simulation with an initial Poisson distribution with λ equal to 1.61. In this simulation, agents of level k incur costs of $k^{1.2}$, and level 0 agents are not capable of maximizing expected utility. The goal of this simulation is to identify a cost function that creates a stable distribution of k-level agents resembling real-world conditions. As discussed in the introduction, various reputable empirical studies suggest that higher-order thinking capacities in humans are currently Poisson-distributed with λ = 1.61. Looking at Figure 4a, the cost function[10] $c(k) = k^{1.2}$ appears to create or explain such a stable equilibrium. However, Figure 4b reveals that this equilibrium may not hold over the long term: while the numbers of level 1 and level 4 agents remain relatively constant, level 3 agents seem to outcompete level 2 and level 0 agents. It is likely that eventually, level 3 agents will completely replace all level 3-x agents,

[10]The costs are calculated per game. The agents can earn between 0 and 29 utilities in each game, from which the respective costs are deducted.

leading to a population composed only of level 3 and level 4 agents. Furthermore, we have hypothesized that in a population with only level 3 and level 4 agents, level 3 agents would eventually go extinct over the long term. This suggests that, ultimately, only level 4 agents would survive. To test this concern, I reran the simulation with the same parameters but extended it to 300 generations instead of 30.



(a) (b)

Figure 5: Poissondistribution (λ=1.5); $c(k) = k^{1.2}$; Level 0 not clever; 300 Generations

The figures 5a and 5b show that level 3 agents initially exhibit rapid growth compared to other agents, which quickly displaces all level 3-x agents. Level 1 agents persist slightly longer, possibly due to their large initial population size. Interestingly, level 2 agents decline faster and go extinct earlier than level 0 agents, despite starting with higher numbers. This provides a counterexample demonstrating that, in a population with a cost function, hypotheses I and II do not hold. However, hypothesis III seems to hold in this scenario: level 4 agents ultimately survive as the sole population. The growth of level 4 agents, though, seems to contradict hypotheses I, IV, and V. In the early stages, level 4 agents perform worse than level 3 agents despite having similar initial numbers. This challenges hypothesis I, which suggests that all else being equal, agents with higher thinking levels should grow faster. Additionally, Hypotheses IV and V support the growth of level 4 agents in the early stages of the simulation because there are more level 4-x than 3-x agents and more 3+x than 4+x agents. However, the simulation reveals that level 3 agents initially grow faster than level 4 agents.

Question 2: Why does the number of level 3 agents grow faster than level 4 agents in an environment with thinking costs, even when $N_4 = N_3$?

Later in the simulation, the growth rates of level 3 and level 4 agents reverse. The growth of level 4 agents accelerates sharply once the population contains fewer level 3-x agents. Between generations 50 and 55, level 3 growth abruptly stagnates—around the time when there are more level 3+x than 3-x agents.

Shortly thereafter, the number of level 4 agents surpasses level 3 agents within a few generations. This shift in reproduction rates between level 3 and level 4 agents can be explained in two ways:

1. Initially, both level 3 and level 4 agents grow, which rapidly reduces the number of 3-x agents while the number of 4-x agents remains relatively stable. According to hypothesis IV, this explains why level 3 growth stagnates while level 4 growth does not.

2. As the number of 3+x agents increases while 4+x agents remain constant, hypothesis V suggests that this also contributes to the shift in growth rates.

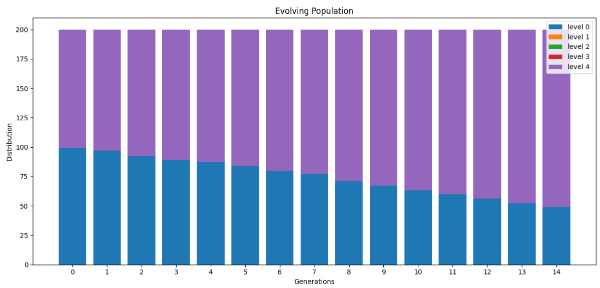
Thus, hypotheses IV and V account for the shift in growth rates between generations 50 and 55. However, a different explanation is needed to address question 2.

Since the growth of level 4 agents seems to depend on the presence of level 3 agents, it is plausible that level 4 agents perform better against level 3 agents than against level 3-x agents. This would explain why level 3 agents initially reproduce more successfully, as they perform better against level 3-x agents.

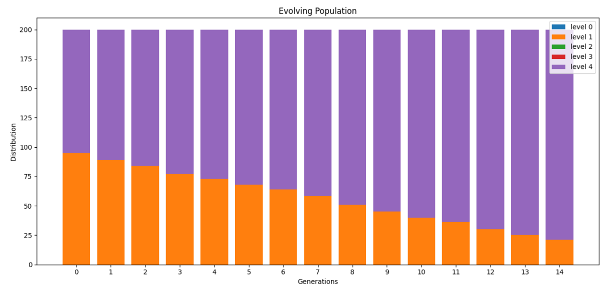
Assumption 1: Level 4 reproduction is more effective against level 3 than level 3-x agents

Assumption 2: Level 3 reproduction against level 3-x agents is higher than level 4 reproduction against level 3-x agents.

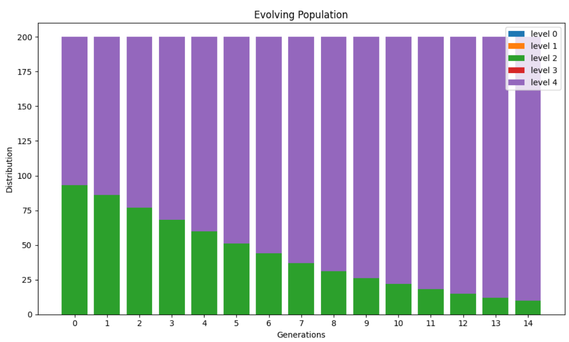
To test the first assumption, a series of simulations with cost functions could be conducted where the first generation consists only of level 4 and level $4 - x_i$ agents (with $x_i \in x$). The ratio of level 4 to level $4 - x_i$ agents in the first generation should be 1:1 across all simulations. The idea is to analyze the growth rate of level 4 agents to determine how well they reproduce against level $4 - x_i$ agents. The faster level 4 agents grow in a population with only level $4 - x_i$ agents, the greater their reproductive success against these agents. Following this, an analysis will be conducted using four simulations, each with 50% level 4 and 50% level $4 - x_i$ agents in the first generation. The same thinking costs ($c(k) = k^{1.2}$) as in the previous simulation will be applied, and the agents will also not exhibit smart behaviors.



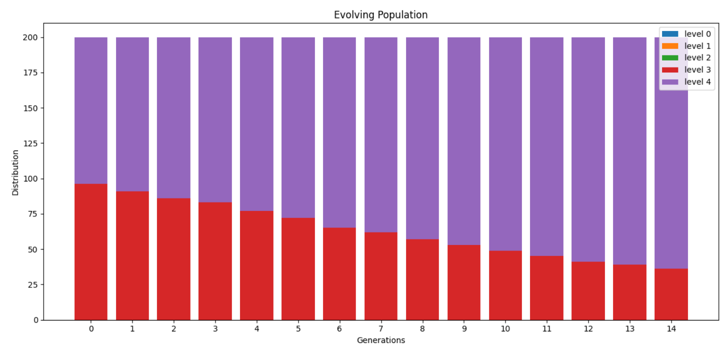
(a) Level 0 vs Level 4



(b) Level 1 vs Level 4



(c) Level 2 vs Level 4



(d) Level 3 vs Level 4

Figure 6: Simulation D: Level 4 vs Level 4-x Agents with costs

From Figure 6, it becomes clear that level 4 agents perform best against level 2 agents, second-best against level 1 agents, third-best against level 3 agents, and worst against level 0 agents. To provide more than just an ordinal ranking and compare the growth of level 4 agents against level 4-x_i agents on an interval scale, the Table 6 includes the average growth (AG10) of level 4 agents in simulation D, as well as their Overall Individual Share (OIS).

Opponent	**AG10 in percent (Level 4)**	**OIS in percent (Level 4)**
Level 3	42.7	68.27
Level 2	60	78.3
Level 1	52.7	71.47
Level 0	34	62.57

Table 6: Performance comparison of AG10 and OIS (Level 4) against different opponents.

From Table 6 it is evident that assumption 1, which suggested that level 4 agents perform better against level 3 agents than against level 3-xi agents, has mostly proven to be incorrect. Both the AG10 and the OIS against level 3 agents are lower than those against level 1 and 2 agents. However, level 4 agents perform worse against level 0 agents than against level 3 agents. To better understand which agent types perform well against which opponents and to test assumption 2, I conducted this ceteris paribus analysis not only with level 4 agents but also with all other levels, and recorded the data in the following table:

Table 7: Average Growth per 10 Generations in percent (AG10)

Agent level / Opponent level	0	1	2	3	4
0	-	-3.3	+16.6	-14.6	-34.0
1	+3.3	-	-10.6	-30.6	-52.7
2	-16.6	+10.6	-	-45.3	-60.0
3	+14.7	+30.7	+45.3	-	-42.7
4	+34.0	+52.7	+60.0	+42.7	-

Table 8: Overall Individual Share (OIS)

Agent level / Opponent level	0	1	2	3	4
0	-	46.76	64.06	41.43	22.26
1	53.23	-	44.56	27.38	17.65
2	35.93	55.43	-	17.3	12.38
3	58.56	72.61	82.26	-	20.28
4	77.3	82.235	87.61	79.71	-

From the tables 7 and 8, it is evident that even with costs for higher-order thinking, level 3 and level 4 agents perform the best. Level 2 agents perform the worst, followed by level 1 agents, while level 0 agents occupy the middle ground. This again confirms that Hypothesis I does not hold in simulations with a cost function. Moreover, this table provides an answer to Question 1, showing the influence that k-x_i and k+x_i agents have on the reproduction of k agents.

Answer 1

Unsurprisingly, level-k agents do not benefit equally when competing against all k-x opponents, nor do they suffer equally when facing all k+x opponents. There is also no simple negative linear relationship between the opponent's level and the reproductive success of the agent. In nearly all cases agents perform best against level-2 opponents, slightly worse against level-1, even worse against level-0, and perform the worst against level-3 and level-4 opponents.

Of course, the OIS and AG10 values must be considered in relation to the distribution in each generation, and the underlying simulation involves costs for higher-order thinking. Nonetheless, a trend can be observed regarding which agents benefit most from the presence of other agent types. Additionally, against our assumption, table 7 shows that Hypotheses IV and V largely hold true in simulations with cost functions: All k agents benefit from agents with lower levels and shrink when facing agents with higher levels, except for the case where level 0 agents benefit from level 2 agents, and level 2 agents shrink against level 0 agents.

Our assumption 2 is contradicted by these measurements: the OIS and AG10 for level 4 agents against all level 3-x_i agents are higher than the OIS and AG10 of level 3 agents against the level 3-x_i agents. So, we still haven´t found an explaination why level 3 agents perform better at the start of simulation C. The fact that level 3 agents perform better means that the sum of the utilities for level 3 agents against all level k agents (including other level 3 agents) is higher than the sum of utilities for level 4 agents against all level k agents. Since, from the tables 7 an 8, we know that level 4 agents perform better against 3-x agents and also perform better against level 4-1 agents than level 3 agents against level 3+1 agents, level 3 agents must perform better against other level 3 agents than level 4 agents against other level 4 agents, as highlighted in table 9 .

Lvl 4 vs Lvl 0	$>$	Lvl 3 vs Lvl 0
Lvl 4 vs Lvl 1	$>$	Lvl 3 vs Lvl 0
Lvl 4 vs Lvl 2	$>$	Lvl 3 vs Lvl 0
Lvl 4 vs Lvl 3	$>$	Lvl 3 vs Lvl 0
Lvl 4 vs Lvl 4	?	Lvl 3 vs Lvl 0
=	=	=
Lvl 4 vs Lvl k	$<$	Lvl 3 vs Lvl 0

Table 9: Performance of Level 4 vs Level 3 Agents

In the appendix, there is a mathematical proof (6)showing that Level 3 agents generate more utility against other Level 3 agents, even without a cost function, compared to Level 4 agents playing against opponents with the same thinking level.

This effect is further amplified in simulations with a cost function, as level 4 agents incur more utility deductions than level 3 agents. Our equation with the cost function is as follows:

$$E(4\,vs\,4) - c(4) \overset{?}{<} E(3\,vs\,3) - c(3) \;\Big|\, (c(k) = k^{1.2}) \rightarrow c(4) - c(3) \sim 1.5$$

$$E(4\,vs\,4) - 1.5 < E(3\,vs\,3)$$

Thus, we have found the answer to question 2:

Answer 2: Level 3 agents perform better against agents of the same level, especially when there are costs for higher-order thinking. This better performance against agents of the same level seems to compensate for the fact that level 4 agents perform better against all other agents at the start of simulation C.
$E(4\,vs\,4) < E(3\,vs\,3)$

This shows us that, despite the initially higher reproductive success of Level 3 agents compared to Level 4 agents in the simulation with costs, Hypotheses III to VI are confirmed, while Hypotheses I and II are not.

The question arises as to how exactly the reproductive success of the agent types has changed compared to Simulation A, which had no costs for higher-order thinking. The difference in OIS (Overall Individual Success) is presented in Table 10

	Level 0	Level 1	Level 2	Level 3	Level 4
Without costs – Simulation A	3.31	10.5	11.9	28.38	45.9
With costs – Simulation C	25.1	28.73	16.62	23.9	5.64
Absolute OIS Difference	21.79	18.23	4.72	-4.48	-40.26
Relative OIS Difference	+658%	173.62%	39.66%	-15.79%	-87.17%

Table 10: OIS comparison (%) of Simulation A and C

It becomes clear that the lower levels (0 to 2) benefit from the costs, while Level 3 and Level 4 agents experience a reduction in their reproductive success. Additionally, it stands out that the lower the agent level, the greater the absolute and relative change in reproductive success. This negative linear correlation between the thinking level and the relative OIS difference is illustrated in the following graph.

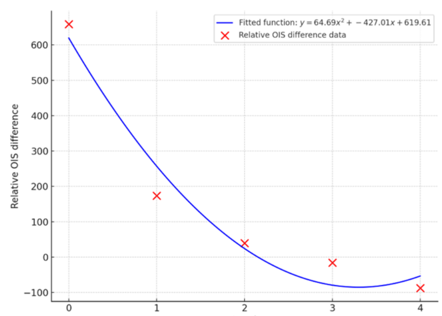
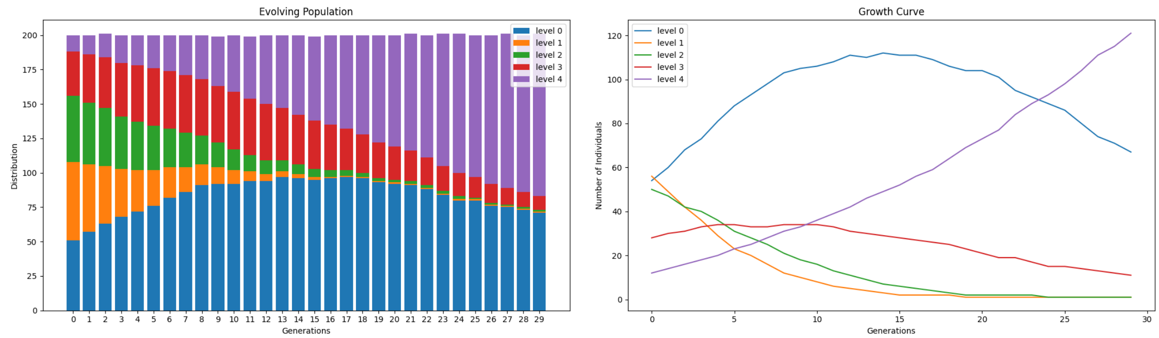


Figure 7: Relative OIS Difference Simulation A and C

The approximated function shows that the relative OIS differences for the lower levels are significantly larger than for the higher levels. This highlights the effect of adding costs on the agents' reproductive success. This strong effect on the low-level agents' reproductive success is partly because they suffer less from the costs. (Reminder: the cost function is $c(k) = k^{1.2}$). On the other hand, they also benefit from a kind of compound interest effect: Since any low-level agent-types generate relatively more utility due to the lower costs in a generation x, they have a larger population share in generation x+1 . This again leads to an even greater utility sum for this agent type in the next generation.

4.6 Simulation E



(a) (b)

Figure 8: Poissondistribution (λ=1.5); c(k)=0; Level 0 is clever

In Simulation E, shown in Figure 8, the first generation is Poisson-distributed with lambda equal to 1.61. There are no costs for higher-order thinking; however, the level 0 agents are "smart" in the sense that, instead of randomly choosing one of three actions, they calculate the expected utility of the three actions and choose the one with the highest expected utility. They assume their opponent plays all three actions equally likely, making all three counteractions equally probable. Notably, their theory of mind is as non-existent as that of standard level 0 agents, i.e., they do not consider what their opponent might do.

In this simulation, the growth ratios of level 1 to level 4 agents are not significantly different from those in Simulation A, with the difference being that the "smart" level 0 (S0) agents rise from last place to second place in terms of performance. More specifically, they have the second-highest growth rate and die last, just before the level 4 agents take over the entire population. This contradicts Hypotheses I and II. It is noteworthy that the smart level 0 agents perform significantly better than all other agents but cannot surpass the level 4 agents. This may be because the level 4 agents are the only ones consistently playing the Nash equilibrium, which cannot be beaten through expected utility maximization. Assumption 3: The frequency of Nash equilibrium actions of agents determines their performance against smart level 0 agents. It remains uncertain whether the level 4 agents would still perform better if they did not exclusively play the Nash equilibrium, such as in a scenario with more than three actions per player in the game matrices.

In order to investigate Assumption 3 we will take a look at the difference of the OIS in comparison to Simulation A without expected utility maximizers[11]. Due to the growth of level 0 agents increasing significantly compared to the simulation with the "dumb" level 0 agents, the growth of other agents has

[11] If Assumption 3 holds, we would expect higher level agents to loose less performance in Simulation E compared to Simulation A, because they play the NE Actions more often.

decreased. A comparison of the Overall Individual Share (OIS) of both simulations reveals how much each agent type suffers in reproductive capability due to level 0 agents becoming expected utility maximizers. Table 11 shows the percentage OIS for all agent classes in both simulations, comparing these values both absolutely and relatively.

	Level 0	Level 1	Level 2	Level 3	Level 4
Normal Level 0 – Simulation A	3.31	10.5	11.9	28.38	45.9
Smart Level 0 – Simulation E	42.97	5.78	6	15.28	29.99
Absolute OIS Difference	39.66	-4.63	-5.9	-13.1	-15.91
Relative OIS Difference	+1198%	-44%	-49%	-46%	-34%

Table 11: OIS comparison (%) of Simulation A and E

The level 1, 2, and 3 agents seem to have lost OIS in a similar manner, while level 4 agents have lost significantly less. If our assumption 3 is correct, it would have been expected that higher-level agents would suffer less in their reproductivity (and thus OIS) than lower-level agents. This is because the higher the thinking level, the more often an agent will play the Nash equilibrium because lower-level agents randomize among many actions, one of which is the Nash action, while higher-level agents randomize among fewer actions, one of which is also the Nash action. However, since Table 11 shows that level 1 to level 3 agents lost the same amount of OIS, assumption 3 is not confirmed by this data.

To better understand the performance of the expected utility maximizer and his opponents, we need to examine the probabilities with which the expected utility maximizer chooses actions.

In the appendix, the mathematical derivation of Table 12 is provided (Proof III on page 47), this table summarizes the distribution with which a S0 agent chooses actions.

	Rowplayer	Columnplayer
O	35.29	-
M	35.29	-
U	29.41	-
L	-	44.44
m	-	0
R	-	55.55

Table 12: Probability of S0 choosing an action(%)

Based on the distribution of pure strategies of the smart Level 0 agents (S0), we can see that they are quite similar to the mixed strategy of the Level 1 agents. The smart Level 0 agents perform just as many steps of iterative elimination of dominated strategies[12]. However, Level 0 agents do something that Level 1 agents do

[12] Although they do not think about what their opponent is doing, this is not necessary for the first step of IESDS, because it never makes sense to play the dominated action m from a utility standpoint, regardless of what the opponent plays.

not: they choose the action with the highest expected utility from the remaining actions, instead of randomly choosing among them as Level 1 agents do. For some reason this results in a probability distribution of their strategies that is very similar to the mixed strategy of Level 1 agents, and therefore also to the probability distribution over the Level 1 agents' actions.

But the key difference that allows the smart Level 0 agents to perform much better than the Level 1 agents is that the distribution of the Level 0 actions is based conditionally on the game-specific utility values. This means that when looking at all the different games, Level 0 agents may choose action X with a similar probability to the Level 1 agents, but the expected utility that a Level 0 agent receives is higher than if another agent (e.g., Level 1) receives the same utility, because the expected utility value for the Level 0 agent conditionally depends on the fact that the sum of all expected utility values of the chosen action is higher than those other actions. As a result, the Level 0 agent performs significantly better, despite having a similar distribution.

For the expected utility values of the opponents of the S0 agents, however, this difference is not relevant: while the expected utility that the Level 0 agent receives is increased by choosing the corresponding action, this does not affect the other utility value of the associated strategy profile.

For example: a Level 0 agent plays as the row player against any opponent, and the resulting strategy profile is (O, M) with utility values (x_3, b_1). The expected value of x3 is increased because the Level 0 agent chose action O, as this means that $\sum x_i > \sum y_i \, and \sum x_i > \sum z_i$. However, the expected value of b_1 is not increased.

Let's summarize:

1. Level 0 agents play in a random game actions with almost the same probability as Level 1 agents.
2. For the expected utility of the opponents, it does not matter whether they end up in a certain strategy profile in a game against a Level 0 or Level 1 agent.
3. The mixed strategies of the 0+x agents are constant and thus independent of the specific utility values and thus also of the strategies of the Level 0 agents.

From 1, 2, and 3, it can be concluded that the expected utility of an agent against Level 1 opponents is very similar to that against S0 opponents. But this does not mean their performance is also similar. As elaborated above the S0 Agent is gathering a lot more utility then the Level 1 agent and therefore increasing the overall utility of all agent types and thus the relative utility share (OIS) of the other agent types.

In summary, the opponents of the S0 agents generate similar absolute utility against the S0 agents as against Level 1 agents but have a reduced reproductive capacity because the S0 agents reduce the relative utility of their opponents, by increasing the utility of all agents. Whether the frequency of NE actions affects the utility against S0 agents can be neither confirmed nor disproved by this analysis.

Due to the increased performance of the Level 0 agents, hypotheses I and II no longer hold. Hypothesis III remains valid in this simulation, while the simulation neither supports nor disproves hypotheses IV and V.

5 Key Results

Let us summarize the findings from the results: Six main hypotheses have emerged:

Hypothesis I: The higher the higher-order thinking level, the better the reproductive ability of an agent.

$N^{g+1}_{k-x} - N^{g}_{k-x} < N^{g+1}_{k} - N^{g}_{k}\; for\, x \in [1, k]$

Hypothesis II: The lower the higher-order thinking level, the faster an agent type goes extinct.

$N^{g'}_{k} = 0 \wedge N^{g''}_{k-x} = 0 \Rightarrow g' > g''\; for\, x \in [1, k]$

Hypothesis III: In the long run, the agent type with the highest thinking level always survives.

$\lim_{g\to\infty} N^{g}_{max(k)} = N$

Hypothesis IV: A larger number of agents with lower thinking levels increases the reproductive success of an agent type (this holds even if population size is not constant).

$N'^{\mathbf{g}}_{k-x} > N''^{\mathbf{g}}_{k-x} \wedge N_k' = N_k'' \Rightarrow N'^{\mathbf{g}+1}_{k} > N''^{\mathbf{g}+1}_{k}$

Hypothesis V: A smaller number of agents with higher thinking levels increases the reproductive success of an agent type (this also holds if population size is not constant).

$N'^{\mathbf{g}}_{k+x} > N''^{\mathbf{g}}_{k+x} \wedge N_k' = N_k'' \Rightarrow N'^{\mathbf{g}+1}_{k} < N''^{\mathbf{g}+1}_{k}$

In simulations where there were no costs for higher-order thinking and no expected utility maximizers, all six hypotheses were confirmed. However, in simulations with costs, Hypotheses I and II were no longer valid. The lower level agents dramatically improved in their reproduction which led to level-0 and 1 agents outperforming and outliving level-2 agents. Hypothesis III, which states that in the long term the agents with the highest level dominate the population, remained valid even in simulations with costs. However, this hindered the fulfillment of the goal to find a cost function that would stabilize the reproductive success and population shares of the agent types in equilibrium. While such an equilibrium seemed to form in the short term, after observing 300 generations, it became clear that only level-4 agents survived in the end. It is possible that other more complex cost functions could lead to such equilibrium, but I was unable to identify them. Despite the long-term dominance of level-4 agents, level-3 agents achieved better growth rates in parts of the simulations, which first seemed to contradict Hypothesis IV and V. But we found that this temporary dominance of level 3 over 4 is in accordance with Hypotheses IV and V. The reason is that level-3 agents' expected higher utility exceeded that of level-4 agents when interacting with an agent of the same kind. This was true even in simulations without costs but was further amplified by thinking costs. Additionally, Table 8 shows the specific effects of k-x and k+x opponents on the reproduction of k-level agents, which revealed that, in simulations with thinking costs, agents performed best against level-2 opponents, slightly worse against level-1, even worse against level-0, and worst against level-3 and level-4 opponents.

In the simulation where level-0 agents became expected utility maximizers, their reproductive ability improved from the worst to the second-best. It was observed that the actions of smart level-0 agents closely resembled those of level-1 agents. This caused the utility generated for level-1+x agents to change as if the level-0 agents had been replaced by level-1 agents. But their performance declined drasticly, because the S0 Agents ability to accumulate high utility decreased their OIS. The high utility of smart level-0 agents comes from their ability to perform the first IESDS (Iterated Elimination of Strictly Dominated Strategies) step and achieve higher expected utility values when they make a move compared to other agents. So for the Simulation with Expected utility maximizers, Hypotheses I and II are wrong, III is true, and IV and V are neither supported nor disproved.

6 Interpretation

This simulation has shown that, without costs for Theory of Mind (ToM), having a higher thinking level increases reproduction. This leads to an "arms race", where agents with higher ToM levels can easily invade populations with lower levels, causing these lower-level agents to quickly go extinct. Over time, this process results in the population's overall ToM capacity increasing.

However, even in a cost-free environment, there is a natural limit for ToM. The simulation revealed that performing more than 4 recursive thinking steps provided no additional benefit because strategies stabilized at a Nash equilibrium. Beyond this limit, agents no longer gained any advantage by increasing their ToM capacity further. Therefore, the arms race is naturally constrained: while agents with higher thinking levels always perform better up to this boundary, there is no benefit to exceeding it. This aligns with Stahl´s (1992)[27]proposition that there is an upperlimit to benefits for cognitive sophistication.

When costs are introduced, this dynamic changes. The simulation showed that agents with lower ToM levels could sometimes outperform agents with higher levels because the costs of additional recursive thinking steps outweighed their benefits. These costs significantly increased the reproductive success of agents with lower ToM capacities, enabling them to outperform agents with medium ToM levels. Thus, ToM costs create an incentive to adopt either a naive strategy (low ToM) or a highly complex strategy (high ToM). In contrast, having medium-level ToM abilities appears to provide fewer advantages when costs are considered.

Nevertheless, as highlighted by Nagel (2019)[22] and other empirical studies, many individuals in human populations seem to possess medium-level ToM capacities. This observation suggests that factors beyond ToM costs might influence reproduction. Alternatively, the real-world cost function for ToM might differ from the one applied in this simulation.

One additional factor revealed by the simulation is the presence of attractive alternatives to high ToM abilities. The impressive performance of naive expected utility maximizers, even in a cost-free ToM environment, demonstrates that EUM strategies can provide serious evolutionary competition to high ToM abilities. This finding supports Lenaerts et al´s (2024)[17] proposition that agents with bounded reasoning can dominate in evolutionary competion by using strategies that balance low thinking costs with strategical efficiency. As Robalino and Robson (2016)[24] put it: there is a trade off between cognitive costs and strategic sophistication, and my simulation raises the possibility that alternatives to high ToM sophistication exist, that make a better trade off in this regard.

In summary, the simulation demonstrates that, in principle, higher ToM levels are advantageous. However, ToM costs have a significant impact on evolutionary success, favoring either very low or very high ToM strategies while disadvantaging medium-level abilities. Additionally, alternative strategies, such as expected utility maximization, represent a strong evolutionary competitor to the development of ToM. This combination of factors helps explain why higher-order thinking abilities do not universally dominate in human populations.

References

[1] P. Battigalli and M. Siniscalchi. Theory of mind and strategic thinking. *Games and Economic Behavior*, 38(2):124–142, 2002.

[2] P. Battigalli and M. Siniscalchi. A theory of rational play in normal-form games. *Games and Economic Behavior*, 44:111–143, 2003.

[3] B. D. Bernheim. Rationalizable strategic behavior. *Econometrica*, 52(4):1007–1028, 1984.

[4] A. Bosch-Domenech et al. Strategic behavior in guessing games: A survey of experimental results. *European Journal of Political Economy*, 18(1):1–14, 2002.

[5] Antoni Bosch-DomÃšnech, JosÃ© G. Montalvo, Rosemarie Nagel, and Albert Satorra. One, two, (three), infinity, ... : Newspaper and lab beauty-contest experiments. *The American Economic Review*, 92(5):1687–1701, 2002.

[6] Martin BrÃŒne and Ute BrÃŒne-Cohrs. Theory of mindâevolution, ontogeny, brain mechanisms and psychopathology. *Neuroscience and biobehavioral reviews*, 30:437–55, 02 2006.

[7] C. F. Camerer, A. Dreber, E. Forsell, T.-H. Ho, J. Huber, M. Johannesson, and H. Wu. Evaluating replicability of laboratory experiments in economics. *Science*, 351(6280):1433–1436, 2013.

[8] C. F. Camerer, T.-H. Ho, and J.-K. Chong. A cognitive hierarchy model of games. *The Quarterly Journal of Economics*, 119(3):861–898, 2004.

[9] J. C. Cohrs. The role of epistemic motivation in social attitudes: Psychological needs underlying prejudice, conservatism, and authoritarianism. *Personality and Social Psychology Bulletin*, 32(5):584–597, 2006.

[10] V. P. Crawford, M. A. Costa-Gomes, and N. Iriberri. Structural models of nonequilibrium strategic thinking: Theory, evidence, and applications. *Journal of Economic Literature*, 51(1):5–62, 2013.

[11] H. de Weerd, R. Verbrugge, and B. Verheij. How much does it help to know what she knows you know? an agent-based simulation study. *Artificial Intelligence*, 258:54–72, 2013.

[12] D. Estes, H. M. Wellman, and P. Walley. Theory of mind and social development. *Journal of Child Psychology and Psychiatry*, 30:401–416, 1989.

[13] T. Feddersen and W. Pesendorfer. The swing voter's curse. *American Economic Review*, 88(5):1271–1288, 1998.

[14] G. S. Halford. *Children's understanding: The development of mental models.* Lawrence Erlbaum Associates, 1993.

[15] E. Kimbrough, N. Robalino, and A. Robson. The evolution of theory of mind: Theory and experiments. *Journal of Economic Theory*, 144:272–290, 2014.

[16] T. Kneeland. Identifying higher-order rationality. *Econometrica*, 83(5):2065–2079, 2015.

[17] Tom Lenaerts, Marco Saponara, Jorge M. Pacheco, and Francisco C. Santos. Evolution of a theory of mind. *iScience*, 27(2):108862, 2024.

[18] Felix Mauersberger, Rosemarie Nagel, and Christoph BÃŒhren. Bounded rationality in keynesian beauty contests: A lesson for central bankers? *Economics E-Journal*, 14:1–20, June 2020.

[19] E. Mohlin. Evolution of theories of mind. *Games and Economic Behavior*, 75(1):299–318, 2012.

[20] J. F. Muth. Rational expectations and the theory of price movements. *Econometrica*, 29(3):315–335, 1961.

[21] R. Nagel. Unraveling in guessing games: An experimental study. *The American Economic Review*, 85(5):1313–1326, 1995.

[22] R. Nagel. Thirty years of guessing games: A history of iteration, confusion, and learning. *Journal of Economic Perspectives*, 33(2):186–207, 2019.

[23] J. Perner. *Understanding the representational mind.* MIT Press, 1991.

[24] Nikolaus Robalino and Arthur Robson. The evolution of strategic sophistication. *The American Economic Review*, 106(4):1046–1072, 2016.

[25] M. Siniscalchi. Modeling belief-dependent preferences. *Journal of Economic Theory*, 106(2):356–391, 2002.

[26] M. Siniscalchi. A behavioral characterization of plausible priors. *Journal of Economic Theory*, 111(2):73–88, 2003.

[27] D. O. Stahl. Evolution of smart-n players. Technical report, Levine's Working Paper Archive, 1992.

[28] D. O. Stahl. Evolution of smart players. *Games and Economic Behavior*, 5(4):604–617, 1993.

[29] I. Tsoukalas. Theory of mind: Towards an evolutionary theory. *Evolutionary Psychological Science*, 4(1):38–66, 2018.

[30] P. Walley. *Statistical reasoning with imprecise probabilities.* Chapman and Hall, 1989.

[31] H. M. Wellman, D. Cross, and J. Watson. Meta-analysis of theory-of-mind development: The truth about false belief. *Child Development*, 72(3):655–684, 2001.

[32] J. M. Wooldridge. *Introductory Econometrics: A Modern Approach.*

[33] J. D. Woolley and H. M. Wellman. Origin and truth: Young children's understanding of imaginary mental representations. *Child Development*, 64(1):1–17, 1993.

Appendix

Proof I - General Game Conditions

$$\forall i : a_i < b_i \tag{1}$$

$$\exists i : c_i < b_i \tag{2}$$

$$y_1 > x_1 \wedge y_1 > z_1 \tag{3}$$

$$x_3 > y_3 \wedge x_3 > z_3 \tag{4}$$

$$z_2 > y_2 \wedge z_2 > x_2 \tag{5}$$

$$z_1 < x_1 \wedge z_3 < x_3 \tag{6}$$

$$y_1 > x_1 \vee y_3 > x_3 \tag{7}$$

$$c_1 < b_1 \wedge c_2 < b_2 \tag{8}$$

$$x_3 > y_3 \tag{9}$$

$$from\ (8)\ and\ (10)\ by\ Konjunktion : c_1 < b_1 \wedge c_2 < b_2 \wedge c_3 = b_3 \tag{10}$$

$$from\ (11) : \forall i : c_i \leq b_i \tag{11}$$

$$from\ (12) : \neg\exists i : c_i > b_i \tag{12}$$

$$from\ (13)\ and\ (2)\ by\ IB : \neg(c_3 = b_3) \tag{13}$$

$$IB : case\ 2 : c_3 < b_3 \tag{14}$$

$$from\ (8)\ and\ (15)\ by\ Konjunktion : c_1 < b_1 \wedge c_2 < b_2 \wedge c_3 < b_3 \tag{15}$$

$$from\ (16)\ by\ UE : \forall i : c_i < b_i \tag{16}$$

$$from\ (17)\ by\ \forall N : \neg\exists i : c_i > b_i \tag{17}$$

$$from\ (17)\ and\ (2)\ by\ IB : \neg(c_3 < b_3) \tag{18}$$

$$from\ (19)\ and\ (14)\ by\ LEM : c_3 > b_3 \tag{19}$$

$$from\ (1)\ by\ UB : a_3 < b_3 \tag{20}$$

$$from\ (20)\ and\ (21)\ by\ TRANS : a_3 < b_3 < c_3 \tag{21}$$

$$from\ (1)\ by\ UB : a_1 < b_1 \tag{22}$$

$$from\ (8)\ by\ SIM1 : c_1 < b_1 \tag{23}$$

$$from\ (24)\ and\ (23)\ by\ Konjunktion : c_1 < b_1 \wedge a_1 < b_1 \tag{24}$$

$$from\ (1)\ by\ UB : a_2 < b_2 \tag{25}$$

$$from\ (8)\ by\ SIM2 : c_2 < b_2 \tag{26}$$

$$from\ (26)\ and\ (27)\ by\ Konjunktion : c_2 < b_2 \wedge a_2 < b_2 \tag{27}$$

$$from\ (6)\ by\ SIM1 : z_1 < x_1 \tag{28}$$

$$from\ (7)\ by\ SIM1 : x_1 < y_1 \tag{29}$$

$$from\ (29)\ and\ (30)\ by\ TRANS : z_1 < x_1 < y_1 \tag{30}$$

Proof II - $E(Level\,3\,vs\,Level\,3) > E(Level\,4\,vs\,Level\,4)$

Since in all simulations the actions of level k-agents and the utility values of strategy pairs are at least ordinally specified as summarized in Figure, we can compare the expected utility of Level 4 Agents when playing against other Level 4 Agents ($E[4/4]$), witht the utility of Level 3 Agents playing against Level 3 Opponents ($E[3/3]$)

$$
\begin{aligned}
&c_1 < b_1 \land a_1 < b_1 \\
&c_2 < b_2 \land a_2 < b_2 \\
&a_3 < b_3 < c_3 \\
&z_1 < x_1 < y_1 \\
&z_2 > y_2 \land z_2 > x_2 \\
&x_3 > y_3 \land x_3 > z_3
\end{aligned}
$$

		Player 2		
		L	m	R
Player 1	O	~~x1,c1~~	~~x2,a1~~	x3,b1
	M	~~y1,c2~~	~~y2,a2~~	~~y3,b2~~
	U	~~z1,c3~~	~~z2,a3~~	~~z3,b3~~

Table 13: Ordinal specification of utility values to ensure IESDS

$$E[4/4] = 0.5 \cdot (x_3 + b_1)$$

$$E[3/3] = 0.25 \cdot (x_3 + y_3 + b_1 + b_2)$$

$$E[4/4] \;</>\; E[3/3]$$

$$0.5 \cdot (x_3 + b_1) \;</>\; 0.25 \cdot (x_3 + y_3 + b_1 + b_2)$$

$$2 \cdot x_3 + 2 \cdot b_1 \;</>\; x_3 + y_3 + b_1 + b_2$$

$$x_3 + b_1 \;</>\; y_3 + b_2$$

Table 13 shows that $y_3 > x_3$, while b_1 and b_2 do not follow a clear ordinal hierarchy. However, b_1 and b_2 are each the highest utility values within their respective hierarchies relative to $c_1, a_1 and c_2, a_2$. Therefore, it can be assumed that b_1 and b_2 should have approximately the same average value across the 20 different games[13]. Based on this, we can conclude:

$$x_3 + b_1 < y_3 + b_2$$

[13] Proven in Proof III 6

Thus:

$$E[4/4] < E[3/3]$$

We can conclude that level 3 agents generally gain higher utility when interacting with other level 3 agents than level 4 agents do when interacting with other level 4 agents.

Proof III - Derivation of Action Distribution S0

To analyze the behavior of the S0 agent, we compare the ordinal utility of actions in two cases: when the S0 agent plays as a row player and when it plays as a column player.

Case 1: Row Player

The expected utilities for each action are calculated as follows:

$$E(\text{Action O}) = \frac{1}{3} \cdot (x_1 + x_2 + x_3)$$

$$E(\text{Action M}) = \frac{1}{3} \cdot (y_1 + y_2 + y_3)$$

$$E(\text{Action U}) = \frac{1}{3} \cdot (z_1 + y_2 + y_3)$$

Since the utility values of the strategy profiles are ordinal, we can also compare the expected utilities of actions ordinally. However, a challenge arises because some utility values are not in an ordinal relationship with each other. The ordinal hierarchy of the general game matrix is structured as follows:

$$c_1 < b_1 \wedge a_1 < b_1$$

$$c_2 < b_2 \wedge a_2 < b2$$

$$a_3 < b_3 < c_3$$

$$z_1 < x_1 < y_1$$

$$z_2 > y_2 \wedge z_2 > x_2$$

$$x_3 > y_3 \wedge x_3 > z_3$$

Utility values are always ranked within three distinct positions. When generating specific game matrices, these utility values are drawn uniformly from a list ranging from 0 to 29. Thus, we can compare the expected values of these utilities by their position within their ordinal hierarchy. For example, x_1 ranks second in its ordinal hierarchy (the second largest value). Similarly, y_2 also ranks second within its distinct hierarchy. This implies that both values are derived by drawing three uniformly distributed values from the range [0, 29] and selecting the second-largest. Given that the simulation creates 30 random game matrices, the average values of x_1 and y_1 converge due to the law of large numbers. Therefore, we can consider them approximately equal.

Mathematical Justification for Equal Expected Values

Consider multiple independent groups of three variables, where values are uniformly drawn from the interval $[0, 29]$ and ranked within each group. Each variable in a group has a specific rank (e.g., smallest (1), middle (2), or largest (3)). Due to the uniform distribution, the k-largest value in each group follows the same distribution function. As the law of large numbers states, the average of a random variable approaches its expected value as the number of trials $N \to \infty$. Thus, the average values of variables with the same rank converge to the same expected value across groups when N is large.

This allows us to rank the average utility values across ordinal hierarchies and compare them on an interval scale. Furthermore, the repeated generation of specific matrices from the general matrix normalizes the distances between ordinal values. Since the values are uniformly distributed, the gaps between the ordinal values are also uniformly distributed. Therefore, the average distances between values converge with repeated draws, making the utility values comparable not only on an ordinal scale but also on a ratio scale. Which will we do,based on the ordinal hierachies, in the following:

Utilities of the Row Player

Action O		Action M		Action U	
x1	2	y1	1	z1	3
x2	2	y2	2	z2	1
x3	1	y3	2	z3	2
sum	5	sum	5	sum	6

Table 14: Row Player utility values with their rank in their ordinal hierachy

From the table, we see that the action U provides the highest expected utility least often, while actions O and M equally often provide the highest expected utility. Both O and M provide the highest expected utility one-sixth more frequently than U. Thus, S0 agents choose actions O, M, and U in the ratio $6 : 6 : 5$. Summarizing, as a column player, S0 agents select:

Action O: 35.29% (6/17), Action M: 35.29% (6/17), Action U: 29.41% (5/17).

Case 2: Column Player

Action O		Action M	Action U	
c1	2	dominated action	b1	1
c2	2		b2	1
c3	1		b3	2
sum	5		sum	4

Table 15: Column Player utility values with their rank in their ordinal hierachy

Since action m is dominated by action R, no utility-maximizing agent will choose m. For the other two actions, we proceed with the same logic as above: Action R is expected to provide the highest utility one-fourth more frequently than action L. Thus, S0 agents choose actions L and R in the ratio $5 : 4$. Summarizing, as a row player, S0 agents select:

Action L: 44.44% (4/9), Action R: 55.55% (5/9).

Python Simulation Source Code

```
from Population import *
from Distribution_determening import new_distribution
from Plot import *
from Poisson_distribution import First_Generation_Distribution
from Game_creater import *
import numpy as np

#population distribution in the first generation
Lambda=1.61
Number_Individuals=200
if True:
    Distribution=First_Generation_Distribution(Number_Individuals, Lambda) #poisson distribution
else:
    Distribution=[0,10,0,10,0]  #manual distribution #manual poisson: [45,58,53,26,18]

Generations= 30
Number_of_Games= 30
Thinking_Costs=True
Smart_Lvl0=True
#Parameter der Kostenfunktion: C(k): a * k**b
a=1.2
b=1

Starting_Distribution=new_distribution(Distribution, Number_Individuals)  #saving starting distribution for calculation of
                                                                    #average growth later

for i in range(Generations):

    Utility_after_each_round = [0, 0, 0, 0, 0]
    Population_this_round = Population(Distribution)
    Game_List=List_of_games(Number_of_Games)

    #everybody plays against everybody, every game

    for index, agent in enumerate(Population_this_round):
        for opponent in Population_this_round[index+1:]:
            for game in Game_List:
```

```
                agent.Play(opponent,game, Utility_after_each_round, Thinking_Costs,a,b, Smart_Lvl0)




    print('Generation: ', str(i))
    print("Utility in generation " , str(i) ,": ", Utility_after_each_round)

    Distribution= new_distribution(Utility_after_each_round, Number_Individuals)
    data_dict(Distribution)

    print("Distribution in generation: " , str(i+1) , ": ", Distribution)
    print("----------------------------------------------------")


    if i == Generations-1:
        Absolute_Growth = np.array(Distribution) - np.array(Starting_Distribution)
        Relative_Growth = Absolute_Growth / np.array(Starting_Distribution)
        Relative_Growth_per_10_Generations = Relative_Growth * (10 / Generations)

        print(Starting_Distribution)
        print("Absolute Growth: ", Absolute_Growth)
        print("Relative Growth in percent:", Relative_Growth)
        print("Relative Growth per 10 Generations: ", Relative_Growth_per_10_Generations)


print("Proportion of all individuals during ", str(Generations)," generations: " , calculate_area_proportions(D))
plot_function(D,Generations)
plot_line_chart(D, Generations)
```

Listing 1: Main simulation

```
def new_distribution(Utility_after_last_round,Sum_of_Individuals):
    Sum=sum(Utility_after_last_round)

    # here the percentile-distribution of level 0, dependent on its former utiliy is determined
```

```
    A0 = Utility_after_last_round[0] / Sum
    A1 = Utility_after_last_round[1] / Sum
    A2 = Utility_after_last_round[2] / Sum
    A3 = Utility_after_last_round[3] / Sum
    A4 = Utility_after_last_round[4] / Sum

    L=[A0,A1,A2,A3,A4]
    L=[round(x * Sum_of_Individuals) for x in L] #rounds and multiplies the percentiles with 100
    #L.append(Sum_of_Individuals-sum(L)) #makes sure that  the values always add up to 100
    return L
```

Listing 2: Replicator dynamics

```
Game1= [
    [[10, 15], [6, 27], [12, 30]],
    [[20, 18], [16, 15], [8, 24]],
    [[6, 15], [18, 9], [10, 12]]
]
import random

def Strategy_rowplayer(rowplayer_level,Game, Smart_Lvl0): #this function defines the strategy of
    the rowplayer it always returns one row of the matrix
    if rowplayer_level==0 and Smart_Lvl0 == False:
        return random.choice(Game)
    if rowplayer_level==0 and Smart_Lvl0 == True:
        return MaxExpUtil_Row(Game)
    if rowplayer_level==1:
        return random.choice(Game)
    if rowplayer_level==2:
        return random.choice(Game[:2])
    if rowplayer_level == 3:
        return random.choice(Game[:2])
    if rowplayer_level == 4:
            return Game[0]



def Strategy_columnplayer(rowplayer_level, columnplayer_level,Game, Smart_Lvl0): #this function
```

```
defines the strategy of the column-player, it always returns the (utility of the) strategy combination of both players
    Row_strategie=Strategy_rowplayer(rowplayer_level,Game, Smart_Lvl0)
    if columnplayer_level==0 and Smart_Lvl0 == False:
        return random.choice(Row_strategie)
    if columnplayer_level==0 and Smart_Lvl0 == True:
        return MaxExpUtil_Column(Game, Row_strategie)
    if columnplayer_level==1:
        return random.choice([Row_strategie[0], Row_strategie[-1]])
    if columnplayer_level==2:
        return random.choice([Row_strategie[0], Row_strategie[-1]])
    if columnplayer_level == 3:
        return Row_strategie[2]
    if columnplayer_level == 4:
        return Row_strategie[2]

#given the levels of the players they play a certain strategy or randomize if they are indifferent between two or more strategies
#these two function first select which strategie the rowplayer will choose and then which strategy the columplayer will choose

#print(Strategy_columnplayer(3,2,Game1))

def MaxExpUtil_Row(Game):
    Util_0=Game[0][0][0]+Game[0][1][0]+Game[0][2][0]
    Util_1=Game[1][0][0]+Game[1][1][0]+Game[1][2][0]
    Util_2=Game[2][0][0]+Game[2][1][0]+Game[2][2][0]

    if Util_0 >= Util_1 and Util_0 >= Util_2:
        return Game[0]

    if Util_1 >= Util_0 and Util_1 >= Util_2:
        return Game[1]

    if Util_2 >= Util_1 and Util_2 >= Util_0:
        return Game[2]


def MaxExpUtil_Column(Game, Row_strategie):
```

```
    Util_0 = Game[0][0][1] + Game[1][0][1] + Game[2][0][1]
    Util_1 = Game[0][1][1] + Game[1][1][1] + Game[2][1][1]
    Util_2 = Game[0][2][1] + Game[1][2][1] + Game[2][2][1]

    if Util_0 >= Util_1 and Util_0 >= Util_2:
        return Row_strategie[0]

    if Util_1 >= Util_0 and Util_1 >= Util_2:
        return Row_strategie[1]

    if Util_2 >= Util_1 and Util_2 >= Util_0:
        return Row_strategie[2]
```

Listing 3: Determine strategy for agents

```
#this files creates a random game which has a structure that allows iterated elimination of
    dominated strategies

import random

def random_game():

    Utilities=range(30)

    while True:
        a1, b1, c1 = random.sample(Utilities,3)

        if a1 < b1 and c1 < b1:
            break

    while True:
        a2, b2, c2 = random.sample(Utilities, 3)

        if a2 < b2 and c2 < b2:
            break

    while True:
        a3, b3, c3 = random.sample(Utilities, 3)
```

```

        if a3 < b3 < c3:
            break

    # now the utility values for the rowplayer

    while True:
        x1, y1, z1 = random.sample(Utilities, 3)

        if z1 < x1 < y1:
            break

    while True:
        x2, y2, z2 = random.sample(Utilities, 3)

        if x2 < z2 and y2 < z2:
            break

    while True:
        x3, y3, z3 = random.sample(Utilities, 3)

        if y3 < x3 and z3 < x3:
            break

    #now we have all the variables for the matrix

    return  [
        [[x1, c1], [x2, a1], [x3, b1]],
        [[y1, c2], [y2, a2], [y3, b2]],
        [[z1, c3], [z2, a3], [z3, b3]]
    ]



def List_of_games(Number_of_Games):
    Game_List=[]

    for i in range(Number_of_Games):
        Game_List.append(random_game())
```

```
    print(Game_List)
    return Game_List
```

Listing 4: Creating the Games

```
import random
from Strategy import *



#Utility_after_each_round=[0,0,0,0]  # here the utility of the individuals will added: the first
    element is the utility of all
                                      # level 0 individuals and so on, this will later determine
                                          the population of the next generation

# Definition of the class for the object
class Individual:
    def __init__(self, level):
        self.level = level

    def get_level(self):
        return self.level

    def cost_function(self,a,b):
        k=self.level
        return a*k**b

    def Play(self, opponent, game, Utility_after_each_round,Thinking_Costs,a,b, Smart_Lvl0):

        Utility_of_game = Strategy_columnplayer(self.level,opponent.get_level(), game, Smart_Lvl0)

        Utility_after_each_round[self.level] += Utility_of_game[0]
        Utility_after_each_round[opponent.get_level()] += Utility_of_game[1]

        if Thinking_Costs==True:
            Utility_after_each_round[self.level] -= self.cost_function(a,b)
            Utility_after_each_round[opponent.get_level()]-= opponent.cost_function(a,b)

```

```
        #print(Utility_after_each_round)
        # #if liked you can print out the utility distribution after each play of every pair

def Population(level_counts):
    # Number of objects for each level
    #level_counts = [15, 25, 35, 25]

    # List to store objects
    object_list = []

    # Iterate over the number of objects for each level
    for level, count in enumerate(level_counts):
        # Create the specified number of objects for each level and add them to the list
        for _ in range(count):
            obj = Individual(level)
            object_list.append(obj)

    # Shuffle the list to ensure the objects are not sorted by level
    random.shuffle(object_list)

    # Example: Output the level of each object in the list
    #for obj in object_list:
        #print("Level:", obj.level)

    return object_list
# returns a list with objects - the individuals in the population
#print(Population([25,25,25,25]))
```

Listing 5: Create the Agents

```
import numpy as np

def First_Generation_Distribution(Number_Individuals, lambd):
    # Ziehe x Mal aus der Liste range(0, 3) mit Poisson-Verteilung
    draws = np.random.poisson(lambd, Number_Individuals)

    List_of_levels=draws.tolist()
    print(List_of_levels)
```

```
    Level_4plus=len(List_of_levels)-(List_of_levels.count(0)+List_of_levels.count(1)+List_of_levels.count(2)+L

    return
        [List_of_levels.count(0),List_of_levels.count(1),List_of_levels.count(2),List_of_levels.count(3),Leve
```

Listing 6: Determines the poisson distribution of the first Generation

```
import matplotlib.pyplot as plt

#Data necessary for plotting:

Individuals=["level 0","level 1","level 2","level 3","level 4"]



D={
        "level 0": [],
        "level 1": [],
        "level 2": [],
        "level 3": [],
        "level 4": []
    }
def data_dict(Distribution):

    # this function collects the distribution data in a dictionary in order to use it later for
        plotting
    D["level 0"].append(Distribution[0])
    D["level 1"].append(Distribution[1])
    D["level 2"].append(Distribution[2])
    D["level 3"].append(Distribution[3])
    D["level 4"].append(Distribution[4])

#Now we start plotting
def plot_function(Dict,Generations):

    time_periods = list(range(0,Generations))
    fig, ax = plt.subplots(figsize=(10, 6))

```

```
    # create plot
    bottom = [0] * len(time_periods)
    for individual in Individuals:
        ax.bar(time_periods, Dict[individual], bottom=bottom, label=individual)
        bottom = [bottom[i] + Dict[individual][i] for i in range(len(time_periods))]

    # set a name for title and axis
    ax.set_xlabel('Generations')
    ax.set_ylabel('Distribution')
    ax.set_title('Evolving Population')
    ax.legend()

    # Show diagramm
    plt.xticks(time_periods)
    plt.tight_layout()
    plt.show()

#other diagramm

def plot_line_chart(Dict, Generations):
    time_periods = list(range(Generations))  # X-axis: Generation numbers

    # Create the figure and axis
    fig, ax = plt.subplots(figsize=(10, 6))

    # Loop through each individual level and plot a line for each one
    for individual in Individuals:
        ax.plot(time_periods, Dict[individual], label=individual)

    # Set labels and title for the axes
    ax.set_xlabel('Generations')
    ax.set_ylabel('Number of Individuals')
    ax.set_title('Growth Curve')

    # Add a legend to identify each line by color
    ax.legend()

    # Display the chart
    plt.tight_layout()
```

```
    plt.show()


#Function calculates the area of the individual colours in the sÃ€ulendiagramm

def calculate_area_proportions(Dict):
    # Berechne die Summe aller Werte Ãœber alle Farben und Generationen
    total_area = sum(sum(Dict[individual]) for individual in Dict)

    # Berechne den Anteil fÃœr jede Farbe
    area_proportions = {}
    for individual in Dict:
        individual_area = sum(Dict[individual])
        area_proportions[individual] = individual_area / total_area

    return area_proportions
```

Listing 7: Plots the results