POLITECNICO DI TORINO

Master degree in Physics of Complex Systems

Master's Degree Thesis

How has evolution shaped our decision making? A Neural Network Agent Based Model for the development of heuristics

SUPERVISORS
Prof. Didier SORNETTE
ETH Zürich
Prof. Luca DALL'ASTA
Politecnico di Torino

CO-SUPERVISORS
MSc Giuseppe FERRO
ETH Zürich
Prof. Matteo GIULIANI
Politecnico di Milano

CANDIDATE
Valeria NANOTTI

Academic year 2019/2020

Abstract

This work focuses on understanding how evolutionary forces, in a complex environment, have shaped the human decision making processes. Within the field of choice under uncertainty, it is experimentally observed that people tend to make irrational and/or controversial choices (e.g. Allais Paradox, Ellesberg Paradox), adopting simple heuristics rather than following rational principles established by the expected utility framework. Through this project, we aim to retrieve such observed irrational preferences as an evolutionary emergent phenomenon. Specifically, the environment we live in is extremely complex, often characterized by highly non-linear and time-evolving conditions. Evolutionary forces, both exogenous (environment) and endogenous (group interaction), made our ancestors develop certain heuristics, which everyday help us taking quick and efficient decisions. The hypothesis we would like to prove is that these "mental shortcuts" work optimally in a real-world-complexity scenario, but only sub-optimally in abstract and oversimplified laboratory setups, leading to the emergence of irrational choice patterns and paradoxes. To demonstrate our point, we developed an Agent Based Model, where agents are first faced with different degrees of uncertainty, with the aim of naturally facilitating the development of heuristics and strategies. Once the training phase has ended, they are then confronted with different tasks (scenario with same degree of uncertainty, risky lotteries), to observe if some empirically reported patterns would arise. In order to include intelligence in our model, we chose as agents Artificial Neural Networks, and we then trained the agents by means of a structured evolutionary algorithm, to mimic the actual Darwinian selection. As results, we observed different decision-making attitudes, depending on the degree of risk and uncertainty of the choices faced by the agents, both in the training evolutionary phase and in the out-of-sample analysis. In light of such results, possible extensions of the present work are outlined, in order to gain a deeper understanding of how decision making works.

Acknowledgement

To start with, I would like to thank my supervisor Prof. Didier Sornette for the opportunity to do this thesis in his Chair, and for his willingness to let me work from remote. I really admire his passion for research and his visionary way of doing science: the meetings with him have been every time an occasion for seeing things from another prospective.
Thanks to my co-supervisor, Giuseppe Ferro, for our weekly calls during lock down, for his infinite patience and for his enthusiasm for my thesis topic. Without his suggestions and support I would have never been able to reach the same results.
Thanks to all the people with whom I crossed path during my Master's degree; to the professors behind the degree course of Physics of Complex Systems, for all the given advice and having been good mentors, and to my friends and colleagues, for all the moments (of study and not) spent together.

A big thanks goes to my parents, my grandparents and in general to my family, for constantly believing in me and rejoicing to every little success I have had. Making you proud is one of the greatest satisfactions in my life.

Thanks to my friends for the moments of fun, fundamentals to periodically and healthfully detach the mind from physics and give my best in these years. To Alessandra, Luca, Lorenzo, Fari, Francesca, Aris for our countless aperitifs and dinners together; to Irene and Lorenzo for being there; to Stefano for our Leuven's experience and being a good confidant when needed.
A special thanks to Lorenzo, Paolo and Riccardo, the ones beside me for years and that haven't run away yet; without their emotional support during these years everything would surely have been way harder.
Last but not least, I want to thank my friend and classmate Carola for always being by my side in this university path; she was there in any good or bad moment, supporting and encouraging me when I felt lost and correcting me when needed. She was the sister I have never had: I am very lucky and grateful to have found a friend like her.

I will forever remember with fondness and nostalgia this beautiful chapter of my life. Thank you all.

Valeria

Contents

List of Figures 5

List of Tables 6

1 Introduction 7

2 Decision Theory State of the Art 10
2.1 Expected Value Theory . . . 10
2.1.1 Criticism of Expected Value Theory . . . 11
2.1.1.1 St. Petersburg paradox . . . 11
2.2 Expected Utility Theory (EUT) . . . 12
2.2.1 Criticism of Expected Utility Theory . . . 13
2.2.1.1 Allais Paradox . . . 14
2.2.1.2 Ellesberg Paradox . . . 14
2.3 Beyond Expected Utility Theory . . . 15
2.3.1 Prospect Theory and Cumulative Prospect Theory (CPT) . . . 15
2.3.1.1 Criticism of Cumulative Prospect Theory . . . 16
2.3.2 Rank-Dependent Utility Theory . . . 16
2.3.3 Quantum Decision Theory (QDT) . . . 18
2.4 List of further violations of Expected Utility Theory . . . 19

3 Heuristics in decision making 21
3.0.1 Satisficing and Bounded Rationality . . . 21
3.0.2 Adaptive Toolbox, Ecological Rationality and Smart Heuristics . 22

4 Purpose of the Agent Based Model 25
4.1 Creation of a complex environment . . . 25
4.2 Agents as Neural Networks . . . 26
4.3 Evolutionary Algorithm for training the NNs . . . 27
4.4 Evaluate the agents: Reverse Engineering . . . 29
4.4.1 Methods specifications: EU and RDU . . . 30
4.4.2 Methods evaluation: Maximum Likelihood . . . 33
4.4.3 Methods comparison: Nested Hypothesis Testing . . . 34

5 Inside the Agent Based Model 36
5.1 Create the initial population . . . 36
5.2 Define the complex environment . . . 36

5.2.1 Input Dataset: introduction of uncertainty . . . 37
5.2.2 Uncertainty targeting: different scenarios . . . 38
5.3 Make the agents learn: Evolutionary Algorithm . . . 39
5.3.1 Evaluate the fitness . . . 39
5.3.1.1 Proportional Selection . . . 42
5.3.1.2 Truncation Selection . . . 42
5.3.1.3 Sigmoid-weighted Selection . . . 43
5.3.2 Agents reproduction . . . 44
5.3.3 Agents mutation . . . 44
5.3.3.1 Intra-generational mutation (learning) . . . 45
5.3.3.2 Inter-generational mutation . . . 45
5.3.4 Repeat across generations . . . 46
5.4 Out-of-sample analysis . . . 46
5.4.1 Out-of-sample dataset . . . 46
5.4.2 Different scoring rules to evaluate performances . . . 47
5.4.3 Study the agents behaviour: Reverse Engineering . . . 48
5.5 Summary of the ABM . . . 49
5.5.1 List of parameters of the ABM . . . 49
5.5.2 Flowchart . . . 50

6 Results **52**
6.1 Assess the algorithm: Accuracy Computation . . . 52
6.1.1 Training in different world scenarios:
Accuracy on training and out-of-sample datasets . . . 52
6.1.2 Training with different numbers of choices:
Accuracy on training set . . . 54
6.1.3 Training with different selection rules:
Accuracy on training set . . . 54
6.2 Confront agents with decision theories: Reverse Engineering Analysis . . . 59
6.2.1 Risk aversion with different selection rules . . . 60
6.2.2 Probability distortion with different uncertainty training scenarios 61
6.2.3 Method comparison with different scenarios . . . 65
6.2.4 Limits of the analysis and possible improvements . . . 67

7 Conclusions and possible extensions **68**

Bibliography

List of Figures

2.1 Value function proposed by Tversky and Kahneman 17
2.2 Probability weighting proposed by Tversky and Kahneman 17

4.1 Representation of a simple Neural Network with one only hidden layer . . 28
4.2 Utility function for different values of the parameter α 31
4.3 Probability weighting function 1st type for different values of δ 31
4.4 Probability weighting function 2nd type for different values of κ and γ . . 32
4.5 Probability of choosing lottery B over lottery A 33

5.1 Neural Network used as agents in the ABM 37
5.2 The two probability distribution functions $B(\alpha_1, \beta_1)$ and $B(\alpha_2, \beta_2)$ 40
5.3 The two probability distribution functions $B(\alpha_3, \beta_3)$ and $B(\alpha_4, \beta_4)$ 41
5.4 Mapping payoff-fitness under sigmoid-weighted selection 43
5.5 Representation of wheel selection rule 44

6.1 Accuracy on training set and on test set SU 55
6.2 Accuracy on training set and on test set U2 55
6.3 Accuracy on training set and on test set U1 56
6.4 Accuracy on training set and on test set WU 56
6.5 Accuracy on training set and on test set FL 57
6.6 Accuracy on training set and on test set CE 57
6.7 Accuracy on training set SU with different length 58
6.8 Accuracy on training set FL with different length 58
6.9 Accuracy on the training set for SU world and different selection rules . . 59
6.10 Fitted curves of utility function for CE scenario 62
6.11 Fitted curves of utility function for FL scenario 62
6.12 Fitted curves of utility function for SU scenario 63
6.13 Fit of same agents outcomes to two different probability functions 63
6.14 Probability weighting SU scenario for different sigmoid inflection points . 64
6.15 Probability weighting different scenarios 66
6.16 Probability weighting different scenarios 66

List of Tables

5.1 Different world scenarios creation 39

6.1 Accuracy on training set, out-of-sample same scenario and out-of-sample different scenario for the different worlds (Ngen=500, sigmoid-weighted selection infl=10%, L=500) 53

6.2 Nested hypothesis testing: p-value and best models 67

1. Introduction

The purpose of this thesis is to gain understanding on how evolution, in a complex and uncertain environment, have shaped the human decision making processes.
Decision making is present in nearly everything we do: from daily to more uncommon decisions, our life is driven by the choices we make. Therefore, due to its centrality in our lives, it is not surprising that decision making captures research attention from a wide range of disciplines, starting from cognitive and social psychology to economics, political science, marketing, engineering, philosophy, etc.
Initially, a great part of decision making research was focused on the concept of selecting the optimal decisions, reducing a decision situation essentially to a mathematical optimization problem [Johnson and Busemeyer, 2010]; the most famous theory in this stream is Expected Utility Theory [Morgenstern and Von Neumann, 1953]. These theories have no space for deviation from the best choices: the decision maker will always choose rationally, whatever the circumstances. Obviously, such an approach is too simplistic, indeed it does not take into account the large amount of endogenous and exogenous factors affecting choice and deceiving the decision maker from the optimal move. In addition, humans are famously bad at understanding probabilities, exhibiting a multitude of context dependent biases and distortions. For these reasons, the rational recipe of Expected Utility Theory shows many limitations in accounting how humans perceive probabilities and uncertainty, leading to the notion of bounded rationality [Simon, 1955] and a long list of behavioral biases and fallacies (e.g. [Allais, 1953],[Ellsberg, 1961], etc.).
Many attempts [Ferro and Sornette, 2020] and new theories (among all Cumulative Prospect Theory [Tversky and Kahneman, 1992]) were developed to explain such fallacies, replacing the objective probabilities with subjective probabilities (also called decision weights). These theories are called descriptive decision models, attempting to describe how humans actually make decisions, rather than trying to find best decisions for any situation. Here, the attractiveness of an event is decomposed into the product of subjective probability and subjective value, although empirical evidence indicate that probability weights and utilities are often not separable in the mind of the decision maker [Kadane and Winkler, 1988] [Pruitt and Hoge, 1965]. This approach still retains an expectation principle, but it focuses on which psychological factors must be added to account for observed human decision preferences [Johnson and Busemeyer, 2010]. Theories including the concept of subjective probabilities usually give better results in predicting human decision making; however their biggest criticism is that they are built ad hoc to justify experimental observation: the parameters of the curves characterizing these theories are trivially fitted from experiments, meaning that there is no theoretical explanation for deriving the values and their nature.
In addition, most decision models are built on artificial and oversimplified hypothesis, for

example they describe how a single individual makes decisions, assuming interaction only with the external environment (e.g. choose between two lotteries with different payoffs), but not with other individuals. Paradoxical results are typically reproduced in laboratory settings and in experiments where reality is traded for control, e.g. participants are asked to answer a list of questions isolated from the rest of the world. From a more realistic perspective, it is unlikely that our ancestors had to face similar setups; on the contrary, they had probably experienced both competition and cooperation between individuals, in a fluctuating and uncertain environment.

Therefore, existing decision making theories are not able to explain fully correctly our decision processes, and many researchers are working on developing a new theory. Among them, Professor Sornette, in his Chair of Entrepreneurial Risks (ETH Zurich), is addressing research questions of decision making, analysing the topic from different point of views. Above all, he developed together with the colleague V.I. Yukalov the Quantum Decision Theory [Yukalov and Sornette, 2008]. An important aspect to investigate, which is nowadays not yet explained, is the origin of our "irrational" behaviour in decision making processes: "How and why did our observed decision-making attitudes develop?" So far, in the scientific community, an important factor that has received relatively little attention is evolutionary pressure.

In this work, we suggest that selective dynamics, acting during the evolution of human populations, led to the formation of certain heuristics [Gigerenzer and Todd, 1999], which everyday influence our decision-making process. The key hypothesis of this work is that the observed human deviations from rational expected utility theories derive from evolutionary forces, which gradually affected our decision modules, in order to promote survival and gene transmission in an extremely complex environment, often characterized by highly non-linear and time-evolving conditions.

In other words, evolutionary processes, both exogenous (environment) and endogenous (interaction), may have shaped our decision-making mechanisms to act optimally in an complex real-world scenario, but sub-optimally in overly simplified laboratory tasks. To sum-up, through this project, we would like to retrieve observed irrational preferences as an evolutionary emergent phenomenon. More concretely, we aim to find which are the simplest and most natural ingredients that can lead to the emergence of irrational patterns of decision, combining both an uncertain environment and interaction (e.g. competition and cooperation) between individuals.

To test our hypothesis, we developed an agent based model (ABM), which will describe the evolution of the decision making process. In agent-based modeling, a system is modeled as a collection of autonomous decision-making entities called agents, which individually assess their situation and take decisions. At the simplest level, an ABM consists of a system of agents and the relationships between them; even the simplest model can exhibit complex behavior patterns and provide valuable information about the dynamics of the real-world system that it emulates [Bonabeau, 2002]. Sophisticated ABM can incorporate advanced tools as neural networks, evolutionary algorithms, or other learning techniques to allow realistic learning and adaptation and evolution of the agents, letting unsuspected behaviors to emerge.

As results, we observed different decision-making attitudes, depending on the degree of risk and uncertainty of the choices faced by the agents in the training evolutionary phase.

The structure of the thesis is the following: in the second Chapter, **Decision Theory State of the Art**, we illustrate the actual state of the art in decision-making, explaining the most famous theories and providing some examples where these models fail.
In the third Chapter, **Heuristics in decision making**, we illustrate the concept of heuristics, with a regression on the interesting works of the professor Gerd Gigerenzer.
In the fourth Chapter, **Purpose of the Agent Based Model**, we start to describe our agent based model, explaining its purpose, the choice of using neural networks and an evolutionary algorithm for training them, concluding with an explanation of the Reverse Engineer algorithm, used to compare our agents behaviour to some existing decision making theories.
In the fifth Chapter, **Inside the Agent Based Model**, we go through each step of the model, explaining in details each step of the evolutionary algorithm and its characteristics.
In the fifth Chapter, **Results**, we report the main results of our Neural Network Agent Based Model, to conclude in the sixth Chapter, **Conclusions and possible extensions**, where we summarize the results and mention some possible further developments of the work.

2. Decision Theory State of the Art

The current chapter will cover the State of the Art in decision theory from a behavioral and game-theoretical point of view.
Most theories of decision making assume that any decision can be abstracted and represented as the selection of a single course of action (e.g. a gamble, a lottery) X, described by the value of the possible outcomes that could result from selecting the action $(x_1, x_2, ..., x_N)$, and the associated probability that each outcome would occur if the action was selected, $(p_1, p_2, ..., p_N)$. Following this notation, N is the number of possible outcomes for one single gamble. The single gamble can be also represented as:

$$X = (x_1, p_1; x_2, p_2; ...; x_N, p_N) \tag{2.1}$$

In a decision making problem, giving an ensemble of gambles $W = (X_1, X_2, ..., X_M)$, where M is the number of available alternatives, the player, i.e. decision maker, has to choose the gamble X_i that he believes optimal. To give a first simple example, we can imagine a decision making problem which concerns a choice between two binary lotteries ($N = 2$, $M = 2$) where $X_1 = A$ and $X_2 = B$:

$$A = (x_1^A, p_1^A; x_2^A, p_2^A) \quad vs \quad B = (x_1^B, p_1^B; x_2^B, p_2^B) \tag{2.2}$$

Here, if the decision maker chooses lottery B, he will receive x_1^B with probability p_1^B or x_2^B with probability $p_2^B = 1–p_1^B$.
Decision where the outcome of a lottery is uncertain but the probabilities are known are generally called decisions under risk; on the contrary, situations where probabilities and/or outcomes are unknown are called decisions under ambiguity or uncertainty.

2.1 Expected Value Theory

The simplest and, from a purely statistical point of view, optimal rule for choosing the best gamble from the ensemble W, is selecting the option X that has the highest expected value. Using the notation of (2.1), the expected value $EV(X)$ of the X-th lottery is calculated as:

$$EV(X) = \sum_{i=1}^{N} x_i p_i \tag{2.3}$$

where N is the number of possible outcomes of the lottery X.

2.1.1 Criticism of Expected Value Theory

The EV rule seems reasonable for gambles played repeatedly many times, however, for gambles with high stakes that are only played once, it turns out that this objective may not be so appealing. Bernoulli [Bernoulli, 1954] observed that most people did not make choices in line with the expected value rule when the values were determined with large objective amounts. He proposed that people did not view monetary outcomes objectively, but subjectively. To make things clearer, imagine to take a decision between the following two options (first choice of Allais paradox, explained in Subsection 2.2.1):

(A) A certain outcome valued at €1 million

(B) An uncertain option with an 89% chance of €1 million, a 10% chance of €5 million, and a 1% chance of receiving nothing

The expected value rule, according to Eq. (2.9), suggests that one should take the second option, because

$$EV(B) = €1.39M > €1M = EV(A)$$

However, given this hypothetical choice, the majority of experimental participants selected A even though it has a lower expected value. Presumably, this is due to the fact that the subjective experience of receiving 5 million instead of 1 million is not five times as pleasurable as receiving 1 million instead of nothing. Rather, as wealth increases, the additional value placed on subsequent increments decreases, i.e. the same amount of money does not have the same subjective value to both a miller and a millionaire. Just with this simple example one can deduce two concepts that expected value theory is missing and that will be deepen in the next section:

- Risk aversion, i.e. humans have the tendency of always choosing certainty over risk.
- Diminishing marginal utility, i.e. the marginal utility of a good or service declines as its available supply increases.

Another famous paradox highlighting the importance of distinguishing between short-run or long-run experiments, in order to compute the worth of a gamble, is the St. Petersburg paradox.

2.1.1.1 St. Petersburg paradox

The St Petersburg paradox was first put forward by Nicolaus Bernoulli in 1713 [de Montmort, 1713], but takes its name from the resolution by Daniel Bernoulli, who published his arguments in the Commentaries of the Imperial Academy of Science of Saint Petersburg [Bernoulli, 1738].
This game is a single player game in which a fair coin is tossed at each stage, and, starting with 2 dollars, the value is doubled every time heads appears. The first time tails appears, the game ends and the player wins whatever is in the pot. Hence, mathematically, the player wins 2^k dollars with probability $\frac{1}{2^k}$, where k is a positive integer equal to the number of consecutive head tosses, or equivalently to the number of tosses. The St. Petersburg gamble has infinite expected value (i.e. infinite expected payoff) and considering nothing but the expected value one should therefore play the game at any price if offered

the opportunity. In relation to common experience, this seems to be a paradoxical result. Georges-Louis Leclerc, Comte de Buffon (1701-1788) (famous for his random needle estimation of π) made an empirical test of the problem and found that in 2048 games a total 10,057 dollars were paid, less than an average of less than 5 dollars per game: it turns out that this game was worth only a very small amount to the participants [Nelson, 2013].
As we will explain later, one can solve this paradox computing the worth of the gamble using a rule including the concept of "saturation of happiness", meaning that one become less and less happy as the money prize increase.

2.2 Expected Utility Theory (EUT)

In order to go beyond the criticisms of expected value rule described in the previous section, a new theory was needed; Nicolas Bernoulli described the St. Petersburg paradox in 1713, prompting mathematicians to develop a new theory as a solution. For this reason, in 1738, Nicolas' cousin Daniel Bernoulli, proposed the basis of Expected Utility Theory in [Bernoulli, 1738], while the theory was formalized almost two centuries later by John von Neumann and Oskar Morgenstern, establishing the basis of modern expected utility theory [Morgenstern and Von Neumann, 1953]. Bernoulli claimed that, in order to account for risk aversion, a nonlinear function of utility of outcomes should be used. Bernoulli further proposed that the goal of the gambler was not to maximize his expected gain but instead to maximize the logarithm of his gain: with this assumption, Bernoulli's paper was the first formalization of the concept of marginal utility, which has broad application in economics. Using the words of Bernoulli: "The determination of the value of an item must not be based on the price, but rather on the utility it yields. [...] There is no doubt that a gain of one thousand ducats is more significant to the pauper than to a rich man though both gain the same amount".
Mathematically, this is solved by the explicit introduction of a function that transforms objective value into subjective utility, the so called utility function. The first used basic utility model, suggested by Bernoulli, was the logarithmic function $U(X) = ln(X)$ (known as log utility), where X is gambler's total wealth, and the concepts of risk aversion and diminishing marginal utility of money are built into it.
In economics, utility is the satisfaction or benefit derived by consuming a product; thus the marginal utility of a good or service is the change in the utility from an increase in the consumption of that good or service; the fall in marginal utility as consumption increases is known as diminishing marginal utility. When talking about monetary outcomes, it is assumed that more is better, and so increasing utility functions are adopted. One simple example is the power utility function,

$$U(X) = X^{\alpha} \tag{2.4}$$

The curvature of the utility function plays an important role in determining decision maker's attitude toward risk. The form in 2.4 allows for describing an individual's risk attitudes with a single parameter: when $0 < \alpha < 1$, the utility function is concave and risk-averse behavior is predicted, whereas a convex function predicting risk-seeking behavior emerges if $\alpha > 1$. It is important to stress that X represents not the single outcome of a lottery

x_i but the total possible wealth of the decision maker: $X = W_0 + x_i$, where W_0 is the initial wealth. The fact that decision makers choose according to changes in final wealth, not considering just the gamble frame, is a fundamental feature differentiating generalized expected utility theories from other theories such as Cumulative Prospect Theory, as we will describe in 2.3. According to Expected Utility Theory, one should select the option with the highest expected utility, that for the gamble X in Eq. 2.1:

$$EU(X) = \sum_{i=1}^{N} U(x_i + W_0)p_i \tag{2.5}$$

The assumed concavity of the utility function is enough to characterize risk-aversion. Specifically, there are two types of risk aversion [Schmidt and Zank, 2008], weak and strong; the first one concerns preferences towards risky and risk-less options, while the second one focuses on attitudes towards degrees of risk, i.e. comparison of lotteries with same expected value but different outcome variances (introducing a mean-preserving spread). Expected utility has the advantage to explain both risk aversions in the same way, i.e. by concavity of utility.
Another concept well explained by expected utility theory is stochastic dominance; there are two types of stochastic dominance;

- First-order stochastic dominance, for which a random variable A has first-order stochastic dominance over random variable B if for any outcome x, A gives at least as high a probability of receiving at least x as does B. In terms of expected utility theory this is ensured by the fact that the utility function is an increasing function.

- Second-order stochastic dominance, which is another commonly used type of stochastic dominance, to compare two distributions and measure the risk aversion. For two gambles A and B, gamble A has second-order stochastic dominance over gamble B if the former involves less risk and has at least as high a mean. In order to expected utility to satisfy second order stochastic dominance the requirement is to have an increasing and concave utility functions.

To sum up, Expected Utility, despite its simplicity, is an extremely powerful normative decision theory. Next subsection outlines its limits and how generalized models of choice have been introduced.

2.2.1 Criticism of Expected Utility Theory

Expected Utility is not able to encompass many situations: in empirical applications, a number of violations have been shown to be frequent and systematic. These fallacies are very important, in the sense that have deepened understanding of how people actually decide, with the aim of building more complete and complex decision theories.
The first example that we will examine is the St. Petersburg paradox, explained in Subsection 2.1.1. Here expected utility theory solves it by assuming a concave utility function (for example the log utility model), to take into account the concept of saturation of happiness. This seems reasonable, but what does happen if one constructs a Super St. Petersburg experiment, in which the expected payoff is 2^{2^k}? In this scenario not only the expected

value will be infinite, but also the log utility turns out to be infinite. Hence, the solution by Bernoulli is not completely satisfying, since the lottery can easily be changed in a way such that the paradox reappears; indeed, one just needs to change the game so that it gives even more rapidly increasing payoffs. For any unbounded utility function, one can find a lottery that allows for a variant of the St. Petersburg paradox. This can be solved changing the expected utility function and choosing another particular utility function that provides a non-infinite expected payoff. This discussion raises some reflections: does make sense to modify and impose ad hoc utilities functions for each problem or gamble? Moreover, the utility function associated to the agents are often considered fixed or, in the best scenario, drawn from various possibilities with a certain probability.
Some other example of emergence of EUT violations are the Allais and Ellesberg paradoxes.

2.2.1.1 Allais Paradox

The Allais paradox was first presented by Allais [Allais, 1953]. In this experiment, each agent has to choose one lottery between A1 and A2 together with one lottery between B1 and B2.

- Lottery A1: €1 million with 100% chance.
- Lottery A2: €5 million with 10% chance, €1 million with 89% chance, and 0 with 1% chance.
- Lottery B1: €1 million with 11% chance and €0 with 89% chance.
- Lottery B2: €5 million with 10% chance and €0 with 90% chance.

It was empirically observed that most people would choose both A1 and B2, which is inconsistent with expected utility theory (violation of the independence axiom), for which choice A1 and A2 can be seen as the same choice (as B1 and B2). Such choice pattern can be explained in terms of subjective probability distortion (certainty effect [Tversky and Kahneman, 1981]), where people tend to underweight high probabilities, considering almost-certain events less likely to happen.

2.2.1.2 Ellesberg Paradox

The Ellsberg's paradox was developed by Daniel Ellsberg [Ellsberg, 1961]. In this experiment, an individual is told that an urn contains 90 balls from which 30 are known to be red and the remaining 60 are either black or yellow. He is asked to choose between the following gambles:

- Gamble A: winning €100 if the ball is red
- Gamble B: winning €100 if the ball is black

And one between the following:

- Gamble C: winning €100 if the ball is not black

- Gamble D: winning €100 if the ball is not red

In most cases people will choose A over B and D over C. Betting for or against the known information (red ball) is perceived safer than betting for or against the unknown (black ball). Nevertheless, these choices of preferences result in a violation of the expected utility theory, which would require the ordering of A to B to be preserved in C to D.
This phenomenon is called ambiguity aversion: Ellsberg showed that not only do people display aversion to risk, they also display aversion to ambiguity, i.e. people tend to prefer gambles for which they are confident and know the exact probabilities involved.

2.3 Beyond Expected Utility Theory

To account for the paradoxes above outlined, several generalized expected utility theories were formalized. Prominent examples are Rank-dependent expected utility and (Cumulative) prospect theory [Johnson and Busemeyer, 2010].

2.3.1 Prospect Theory and Cumulative Prospect Theory (CPT)

Prospect theory was first introduced by D. Kahneman and A. Tversky in 1979 [Kahneman and Tversky, 1979]. It presents a substantial difference from expect utility theory, postulating that carriers of value are not final assets, but gains and losses. In other words, people do not consider their present wealth when making a choice, but they consider only the gamble frame. Thanks to this feature, many psychological mechanisms can be described, as loss aversion (people's tendency to avoid losses rather than acquiring equivalent gains: it is better to not lose €x than to win €x). In 1992 [Tversky and Kahneman, 1992], by incorporating the rank-dependent weighting, improved the theory such that no violations of stochastic dominance were predicted, developing the so called Cumulative Prospect Theory (CPT). Under Cumulative Prospect Theory, the utility of the gamble X is evaluated as

$$U_{CPT}(X) = \sum_{i=1}^{N} v(x_i)\pi_i \tag{2.6}$$

where π_i are the decision weights and $v(\cdot)$, the value function proposed by Kahneman and Tversky [Tversky and Kahneman, 1992], is:

$$v(x) = \begin{cases} x^{\alpha} & if \quad x \geq 0 \\ -\lambda(-x^{\alpha}) & else \end{cases} \tag{2.7}$$

represented in 2.2. The authors estimate $\alpha = 0.88$ and $\lambda = 2.25$ from experimental data. This formulation illustrates the four principal elements of prospect theory:

1. **Reference dependence**: as already introduced, in prospect theory, people derive utility from gains and losses measured relative to some reference point, rather than from absolute levels of wealth: the argument of $v(\cdot)$ is x_i not $W+x_i$. Kahneman and Tversky motivate this assumption, known as "reference dependence," with explicit experimental evidence and by noting that our perceptual system works in a similar way (we are more attuned to changes in attributes such as brightness, temperature than we are to their absolute magnitudes).

2. **Loss aversion**: the value function $v(\cdot)$ is defined in order to captures loss aversion, the idea that people are much more sensitive to losses than to gains of the same magnitude. Mathematically, loss aversion is generated by making the value function steeper in the region of losses than in the region of gains, as shown in Figure 2.1. Kahneman and Tversky infer loss aversion from the fact that most people turn down gambles like (–€100, 0.5 ; €110,0.5).

3. **Diminishing sensitivity**: as shown in Figure 2.1, the value function is concave in the region of gains but convex in the region of losses. This element of prospect theory is known as diminishing sensitivity and it implies that, while replacing a €100 gain with a €200 gain has a significant utility impact, replacing a €1,000 gain with a €1,100 gain has a smaller impact (the same for losses). The concavity over gains is a sign of risk aversion, as in Expected Utility Theory. On the contrary, was experimentally demonstrated that people tend to be risk seeking over losses and this motivates the convexity over losses.

4. **Probability weighting**:In prospect theory, people do not weight outcomes by their objective probabilities p_i but rather by decision weights π_i . The decision weights are computed with the help of a probability weighting function w(·) whose argument is an objective probability. One possible form, motivated by numerous evidence [Gonzalez and Wu, 1999], is:

$$w(p) = \frac{p^\delta}{(p^\delta + (1-p)^\delta)^{1/\delta}} \tag{2.8}$$

   where δ= 0.65 is the value estimated by the authors from experimental data, while the curve is shown in Figure 2.2. The weighting function represents over-weighting of small probabilities and under-weighting of high probabilities.

The difference between cumulative prospect theory and the original version of prospect theory is that the weighting function is applied to the cumulative probability distribution function. This section was written with the help of the reviewing article of Cumulative Prospect Theory written by Barberis, [Barberis, 2013].

2.3.1.1 Criticism of Cumulative Prospect Theory

Even though prospect theory describes more accurately how people evaluate risk in experimental settings, its use is not widespread in orthodox economics. One possible motivation, as discussed in [Barberis, 2013], is that, when considering as carriers of utility are gains and losses, it is often unclear what a gain or loss represents in any given situation, especially in finance.

2.3.2 Rank-Dependent Utility Theory

Quiggin [Quiggin, 1982] suggested a generalization of the expected utility model, the so-called Rank-Dependent Utility Theory (RDU). Its crucial point is to relax the assumption that the utility of a gamble has to be linear in the probabilities. This is justified by the fact that individuals tend to substitute 'decision weights' for probabilities, as discussed in

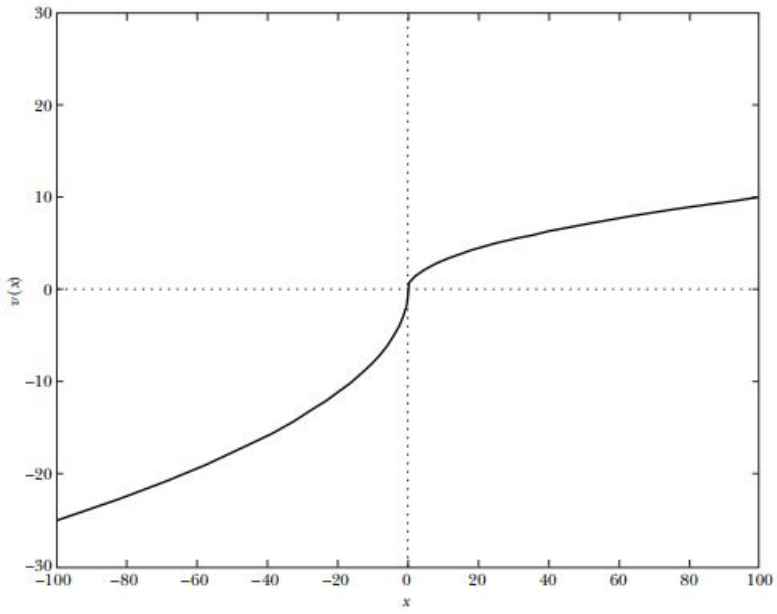


Figure 2.1: Value function proposed by [Tversky and Kahneman, 1992], namely $v(x) = x^{\alpha}$ for $x \geq 0$ and $v(x) = -\lambda(-x)^{\alpha}$ for $x < 0$, where x is an unitary gain or loss. The plot uses $\alpha = 0.5$ and $\lambda = 2.5$, to make loss aversion and diminishing sensitivity easier to see.

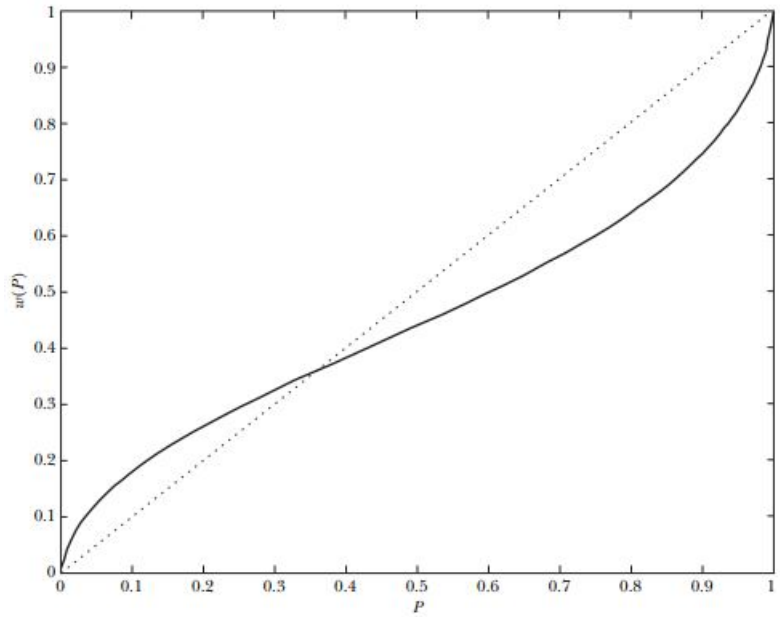


Figure 2.2: Probability weighting function proposed by [Tversky and Kahneman, 1992], namely $w(p) = p^{\delta}/(p^{\delta} + (1-p)^{\delta})^{1/\delta}$, where p is the objective probability. The solid line corresponds to $\delta = 0.65$ while the dotted line corresponds to $\delta = 1$, i.e. linear probability weighting.

previous subsection.
Due to rank-dependence, the weight of an outcome will be not only function of its probability, but also of the ranking position of this particular outcome compared to the others. Indeed, according to RDU, the utility given to an outcome depends not only on its probability but also on the preferability of the considered outcome in comparison to the other possible outcomes.
Mathematically, the utility of an option X becomes:

$$E_{RDU}(X) = \sum_{i=1}^{n} U(x_i + W_0)\pi_i(\vec{p}) \tag{2.9}$$

where W_0 is the initial wealth of the decision maker and $\pi_i(\vec{(p)})$ is introduced to take into account the ranking position of outcome x_i:

$$\pi_i(\vec{p}) = w(\sum_{j=i}^{n} p_j) - w(\sum_{j=i+1}^{n} p_j) \tag{2.10}$$

Here $w(\cdot)$ is the the probability weighting function, which have the shape shown in Figure 2.2 and the same reasoning already explained above for Cumulative Prospect theory.

2.3.3 Quantum Decision Theory (QDT)

Also Professor Didier Sornette is exploring the opportunity to develop a new decision making model, addressing the problem from different point of views, in his Chair of Entrepreneurial Risks at ETH Zurich.
Above all, he developed together with the colleague V.I. Yukalov the Quantum Decision Theory [Yukalov and Sornette, 2008], which provides an intrinsic probabilistic framework describing entangled decision making and non-commutativity of decisions. This theory of decision making takes its name from the fact it is based on the mathematical theory of the Hilbert spaces and on the employment of mathematical techniques used in the quantum theory of physical measurements. The model is led by the natural hypothesis that probabilities and utilities are entangled dual characteristics of the real human decision making process. Indeed, according to QDT, the probability of choosing a prospect can be decomposed into the sum of two terms: the utility factor and the attraction factor; the utility term represents the rational comparison among the available alternatives, while the attraction factor quantifies the attractiveness of a prospect, dependent on interfering feelings, beliefs and subconscious biases. This approach can be seen as a simple mathematical and natural extension of objective probabilities into nonlinear subjective probabilities.
In [Yukalov and Sornette, 2008] the authors demonstrate that the developed quantum formalism allows to explain quantitatively the known anomalies and paradoxes, documented in the context of classical decision theory, without adjustable and ad hoc parameters as in others EUT-substitute theories. As shown in [Yukalov and Sornette, 2011] [Kovalenko and Sornette, 2018], Quantum Decision Theory avoids common paradoxes arising in classical decision theories such as violation of the the conjunction fallacy [Tversky and Kahneman, 1974] (see next Section 2.4 for details) and the already explained Ellsberg's Paradox [Ellsberg, 1961].

2.4 List of further violations of Expected Utility Theory

Here we report a non exhaustive list of further violations and paradoxes of Expected Utility Theory. Investigating the origin of such irrational behaviours is the goal of the present work.

- **Conjunction fallacy**: Formal fallacy that occurs when it is assumed that specific conditions are more probable than a single general one. The most often-cited example of this fallacy originated with Amos Tversky and Daniel Kahneman [Tversky and Kahneman, 1974] and it is called the Linda problem: Linda is 31 years old, single, outspoken, and very bright. She majored in philosophy. As a student, she was deeply concerned with issues of discrimination and social justice, and also participated in anti-nuclear demonstrations. Which is more probable?

  1. Linda is a bank teller.
  2. Linda is a bank teller and is active in the feminist movement.

  The majority of those asked chose option 2. However, the probability of two events occurring together (in "conjunction") is always less than or equal to the probability of either one occurring alone.

- **Preference reversal**: Subjects tend to value p bets (lotteries with a high chance of winning a low prize) lower than r bets (lotteries with a small chance of winning a large prize). When subjects are asked which lotteries they prefer in direct comparison, however, they frequently prefer the p bets over r bets. Preference reversals were initially demonstrated by Lichtenstein and Slovic in 1971 [Lichtenstein and Slovic, 1971].

- **Reflection effect**: It refers to have opposite preferences for gambles differing in the sign of the outcomes. For example, between
  A) Would you prefer to receive €3000 for sure, or €4000 with probability .8?
  B) Would you prefer to lose €3000 for sure, or lose €4000 with probability .8?
  Most people would be risk-averse in question A, and risk-seeking in question B. This issue was solved in prospect theory by Daniel Kahneman and Amos Tversky [Tversky and Kahneman, 1992].

- **Framing effects**: People tend to avoid risk when a positive frame is presented but seek risks when a negative frame is presented. Shown by Tversky and Kahneman in 1981 [Tversky and Kahneman, 1981], with a two-surveys experiment called the Asian disease problem. In both survey scenarios, a group of 600 people is about to be exposed to a deadly disease. In the first survey, the choice is whether to save 200 people for sure, or save all 600 with probability 1/3 (and save nobody with probability 2/3). In the second survey, the choice is whether to allow 400 people to die for sure, or accept a 2/3 probability of all 600 people dying (and a 1/3 probability of nobody dying). The vast majority of participants chose the sure option in the first survey and the gamble in the second, despite they are asking the same thing.

- **Endowment effect**: It refers to the fact that the loss in utility from giving up an item is greater than the gain in utility from acquiring it. It was demonstrated experimentally by Kahneman, Knetsch, and Thaler in 1990 [Kahneman et al., 1990], with the mug experiment: they donate to a percentage of the participants a cheap decorative mug, and asked all subjects to place a monetary value on it. They found that the subjects who were given the mug placed much higher values on it that those who were not, demonstrating that people tend to place much higher values on items they possess, or are "endowed" with.

- **Status quo bias**: It can be intended as the preference for remaining in the current situation. Discussed by William Samuelson and Richard Zeckhauser in 1988 [Samuelson and Zeckhauser, 1988], where they conducted an experiment in which several different funds were described to the participant. When participants were told that they currently had money invested in one of the funds, then they generally preferred that fund to the others. When they were given a neutral scenario in which they did not have money invested in any of the funds, then no such bias was observed.

3. Heuristics in decision making

After having introduced some of the most famous microeconomics theories in Chapter 2, in this Chapter we will analyse decision making from an other point of view, focusing on how heuristics affects our behaviour and how evolution played a first role in the formation of such heuristics.
Heuristics can be defined as mental shortcut or rule-of-thumb strategies allowing people to solve problems and make judgments quickly and efficiently. Indeed, in order to cope with the tremendous amount of information we encounter and to speed up the decision-making process, the brain relies on these mental strategies to simplify situations. In this way we do not have to spend endless amounts of time in analyzing every detail; the number of decisions we make every day is enormous; heuristics allow us to think through the possible outcomes quickly and arrive at a solution.
However, while heuristics speed up the decision-making process, they can introduce errors: these short-cut often lead to inaccurate judgments, and in this respect some paradoxes can arise, especially when the decision maker is forced to take choices in an environment different from the one he was used to [Cherry, 2020].

3.0.1 Satisficing and Bounded Rationality

During the 1950s the Nobel-prize winning psychologist Herbert Simon suggested that human judgment is subject to cognitive limitations, hence it is not logical to search a way to make rational choices. He intended that in real world, people, while taking choices, are limited by the amount of time and information they have at disposal. With this idea, Simon formulated one of the first models of heuristics, known as satisficing, in his 1947 book Administrative Behavior [Simon, 1947].
According to this model, Decision-makers act as satisficers, seeking a satisfactory solution rather than an optimal one and looking through the available alternatives just until an acceptability threshold is met. Simon is also known as the father of bounded rationality, the idea that rationality is limited by intrinsic environmental factors. Bounded rationality complements "rationality as optimization", which views decision-making as a fully rational process of finding an optimal choice given the information available.
Simon's goal was to replace the global rationality of economic man with a kind of rational behavior that is compatible with the access to information and the computational capacities that are actually possessed by organisms, including man, in the kinds of environments in which such organisms exist [Simon, 1955]. Such concept is well represented by the famous scissors analogy: "Human rational behavior (and the rational behavior of all physical symbol systems) is shaped by a scissors whose two blades are the structure of task environments and the computational capabilities of the actor" [Simon, 1990], illustrating

how minds compensate for limited resources by exploiting known structural regularity in the environment.
To sum up, Simon was the first highlighting the big fallacy of existing decision making theories, i.e. assuming an environment characterized by fixed and known probabilities and/or outcomes. This is clearly not the case in everyday life: our world is full of uncertainty (the set of future states and their consequences is not known or knowable) and many real-world problems are characterized by computational intractability or lack of information, both of which preclude the use of mathematical optimization procedures.
Simon urged economists to move away from theories as expected utility models and study how people actually make decisions in realistic situations of extreme time-dependent ambiguity/uncertainty.

3.0.2 Adaptive Toolbox, Ecological Rationality and Smart Heuristics

The thesis of Herbert Simon was revived in the 1990s by Gerd Gigerenzer and others [Gigerenzer and Todd, 1999]. Gerd Gigerenzer proposed a completely new methodology to understand human behaviour in decision making. Gigerenzer was quite a visionary, with his own words: "My work will, I hope, change the way people think about human rationality. Human rationality cannot be understood, I argue, by the ideals of omniscience and optimization. In an uncertain world, there is no optimal solution known for most interesting and urgent problems. When human behavior fails to meet these Olympian expectations, many psychologists conclude that the mind is doomed to irrationality. These are the two dominant views today, and neither extreme of hyper-rationality or irrationality captures the essence of human reasoning. My aim is not so much to criticize the status quo, but rather to provide a viable alternative" [Gigerenzer and Brockman, 2003].
Among researchers, a variety of deviations from rational choice theory were interpreted as systematic flaws in the human mind rather than in the theory [Gigerenzer, 2018]. Mathematicians usually explained the deviations from rationality by modifying models and adding free parameters to decision making theories. But the fundamental problem highlighted by Gigerenzer was that no one can be rational in a world where knowledge is limited and time is pressing. However, traditional models of unbounded rationality and optimization in cognitive science, economics, and animal behavior have tended to view decision-makers as possessing supernatural powers of reason, limitless knowledge, and endless time [Todd and Gigerenzer, 2000].
The two main questions addressed by the Gigerenzer research group were [Gigerenzer and Gaissmaier, 2011]:

1. "Which heuristics do people use in which situations?", analysed in the study of the **Adaptive Toolbox**, where they look for common building blocks (e.g., rules for search) and core capacities (e.g., recognition memory), from which the various heuristics are constructed as an organizing principle. This allows reducing the huge number of heuristics to a smaller number of components.

2. "When should people rely on a given heuristic rather than a complex strategy to make better judgments?", analysed in the study of **Ecological Rationality**, which investigates in which environments a given strategy is better (because there are no best strategies in complex scenarios) than other strategies. The study of ecological

rationality thus involves analyzing the structure of environments together with the structure of heuristics, and then the match between them.

But where do these heuristics come from and what is the role of evolution in their formation? As introduced, each individual or species is assumed to have an adaptive toolbox, i.e. a collection of building blocks and core mental capacities at its disposal for constructing heuristics. Heuristics can be fast and frugal thanks to the fact that these instruments are already in place.
But how is decided and learnt which strategy is the good one to select? First, heuristics and their underlying core capacities are shaped by evolution and by individual learning. Moreover, heuristics are selected and learned by social processes, as in imitation and explicit teaching of heuristics. Finally, the content of individual memory determines in the first place which heuristics are exploited to make decisions [Gigerenzer and Gaissmaier, 2011].
However, what I find to be the most revolutionary concept explored by Gigerenzer is the study of fast and frugal decision-making, or **Smart Heuristics**. Being heuristics cognitive processes, conscious or unconscious, which ignore part of information and save effort, a straightforward conclusion may be that decisions taken thanks to heuristics imply greater errors than rational decisions, defined by optimal-solution models [Gigerenzer and Gaissmaier, 2011].
On the contrary, Gigerenzer discovered that heuristics can lead to fast, frugal, and more accurate than biased decisions in many real-world situations. In other words, in order to make good decisions in an uncertain world, one sometimes has to ignore information: "The art is knowing what one doesn't have to know". [Gigerenzer and Brockman, 2003].
In "Simple heuristics that make us smart" [Gigerenzer and Todd, 1999], fast and frugal heuristics were explored. They discovered that these types of heuristics can enable both living organisms and artificial systems to make smart choices quickly and with a minimum of information by exploiting the way that information is structured in particular environments. These simple heuristics perform comparably to more complex algorithms, particularly when generalizing to new data, i.e. simplicity leads to robustness.
I will conclude with the following bullet points, from [Gigerenzer and Gaissmaier, 2011], to highlight and summarize the most important concepts of the theory of heuristics:

- Heuristics can be more accurate than more complex strategies even though they process less information (less-is-more effects).
- A heuristic is not good or bad, rational or irrational; its accuracy depends on the structure of the environment (ecological rationality).
- Heuristics are embodied and situated, they exploit core capacities of the brain and their success depends on the structure of the environment. They provide an alternative to stable traits, attitudes, preferences, and other internal explanations of behavior.
- With sufficient experience, people learn to select proper heuristics from their adaptive toolbox.
- Usually, the same heuristic can be used both consciously and unconsciously, for inferences and preferences, and underlies social as well as nonsocial intelligence.

- Decision making in organizations typically involves heuristics because the conditions for rational models rarely hold in an uncertain world.

4. Purpose of the Agent Based Model

As mentioned in Chapter 2, when individuals are faced with some kind of choices, they tend to make irrational and illogical decisions. For this reason, many researchers and scientists developed in the years enumerable theories struggling to explain and/or predict human decision making.
Our hypothesis is that the presence of irrationality is due to the action of certain heuristics, i.e. mental shortcuts that allows individuals to make a decision or solve a problem quickly with a minimal mental effort. The complex and uncertain environment has led our brain to elaborate information and take decision using heuristics, which work really well in real world scenarios. However, how will these heuristics perform in oversimplified and unrealistic scenarios? Our hypothesis is that, when applying heuristics to laboratory-situation, they may bring paradoxical results, as indeed observed empirically.
As a tool to test our idea, we develop an Agent Based Model (ABM), representing the evolution of the decision making process from our ancestor to the present. In agent-based modeling, a system is represented as a collection of autonomous decision-making entities called agents, which individually assess their situation and take decisions. At the simplest level, an ABM consists of a system of agents and the relationships between them. Even the simplest model can exhibit complex behavior patterns and provide valuable information about the dynamics of the real-world system that it emulates [Bonabeau, 2002]. Sophisticated ABM can incorporate advanced tools as neural networks, evolutionary algorithms, or other learning techniques to allow realistic learning and adaptation and evolution of the agents, letting unsuspected behaviors to emerge [Bonabeau, 2002]. In this work, we will insert in our ABM both neural networks and an evolutionary algorithm, making it an advanced and sophisticated model.
The idea behind our Agent Based Model is to train intelligent agents in real-world complexity scenarios and let the population evolve under this uncertain environment. After the training part, our aim is to analyse the behaviour of the trained population in specific simplified scenarios, obliging the population to face unrealistic situation to check if paradoxes will appear. Put differently, we want the population to develop a collective brain, with rules and heuristics, and then check if some biases and paradoxes will arise in overly simplified choices.

4.1 Creation of a complex environment

In order to reproduce a real-world scenario, lotteries (the proxy for choices) must change over time, being an exogenous variability a more realistic setup to elicit preference relations. In real world, one does not know the exact probabilities; uncertainty will be always

present and the decision tasks will be much harder with respect to laboratory ones. The only thing that in reality can be achieved is an estimation of expectation; for example, in hunting, one can be able to distinguish between mammals (so at first order we can assume that the payoffs are known with certainty), while the probabilities to capture them are unknown.
We aim to introduce and tune uncertainty through:

- Uncertain probabilities
- Uncertain outcomes
- A combination of both

Our purpose is to develop a model permitting to adjust these values easily, in order to quickly analyse how different levels of uncertainty would affect the results in a decision making process.

4.2 Agents as Neural Networks

If the environment changes over time, we need the agents to be able to generalize their choice criterion, as a classifier does in machine-learning algorithms. Therefore, we have to endow the agents with some generalization ability.
Our purpose is to induce in the system the concept of swarm intelligence [Kennedy, 2006], defined as the collective behavior of natural or artificial decentralized and self-organized systems. Rather than finding the optimal solution by minimization of a complex cost function, one lets microscopic agents follow simple rules, such that an aggregate system develops intelligence as an emergent property. In other words, to optimize a function in a system, it is often sufficient to search for the set of local rules that the system has to follow to make the dynamics converge to a fixed point; it is proven that this fixed point will be the optimal solution, with respect to the imposed cost function [Kennedy, 2006].
The concept of swarm intelligence can be implemented in our agent based model designing a "soup of little brains" evolving, i.e. brains selected and shaped by evolutionary pressure. These brains should be seen as a black box, in the sense that the rules used to make decisions are not a priori imposed by any decision making theory. Then, as soon as the system evolves, one can check the emergence of some similarities with theoretical predictions (for example by fitting existing decision models to the black box synthetic choice data).
We decided to include intelligence in our model by using as agents simple Neural Networks (NNs). Modeled loosely on the human brain, a Neural Network can be seen, in its simplest form, as an unknown black-box performing a non-linear transformation of an input [Hardesty, 2017].
The black-box consists of an ensemble of simple processing nodes that are densely interconnected with predefined rules. These networks are organized into layers of nodes: an input layer and an output layer are always present, and then, located between the input and output of the algorithm, an arbitrary number of hidden layers are included. In the hidden layers, the network applies weights to the inputs and directs them through an activation function as the output; in short, the hidden layers perform nonlinear transformations of the network inputs. The most famous and simple (and the ones that we will implement)

neural networks are called "feed-forward", because data moves in only one direction: a single node might be connected to several nodes in the layer beneath it, from which it receives data, and several nodes in the layer above it, to which it sends data. To each of its incoming connections, a node will assign a number known as a weight. The node receives a different data item over each of its connections and multiplies it by the associated weight. It then adds the resulting products together (plus a number called bias), yielding a single number to pass to a certain activation function. The output of this function will be send along the outgoing connections of the node [Hardesty, 2017].
As an example, we outline the simplest case of feed-forward Artificial Neural Network, represented in Figure 4.1, with one only hidden layer. We can schematize the NN behaviour in the following subsequent steps:

1. The inputs x_i are collected in the nodes of the input layer;

2. Before entering the hidden layer, the inputs x_i are multiplied with the weights $w_{i,j}$ and added to the biases b_j, where j represents the different nodes of the hidden layer;

3. The previous values are then summed in the number inputs, obtaining a single input per hidden layer node;

4. In the hidden layer, this number is passed to the activation function $f(\cdot)$, giving the value a_j;

5. Before entering the output layer and calling k the output node, the values a_j are then combined with the weights $w_{j,k}$ and biases b_k, summed and finally passed to the output layer;

6. The input of the output layer is finally passed to the output function function $g(\cdot)$, resulting in the single output of the NN $o(k)$.

Usually, all weights and thresholds are initially set to random values, and then, during training, they are continually adjusted until the network gives meaningful results.
The usual NNs learning processes follow supervised learning algorithms: given a training set, the NNs are trained with a back-propagation algorithm, inferring a function that maps an input to an output based on example input-output pairs that have been already labeled in advance. For instance, an object recognition system might be fed thousands of labeled images of objects and it would find visual patterns in the images that consistently correlate with particular labels. On the contrary, in our specific case, we want the decision preferences arise from an evolutionary context, and to make the agents learning from their experience. We have therefore to think about another algorithm to train our neural networks.

4.3 Evolutionary Algorithm for training the NNs

As already mentioned, we aim to train the agents in a complex and uncertain environment, to make the agents developing heuristics and learning to take decisions just with intrinsic survival and huge uncertainty.

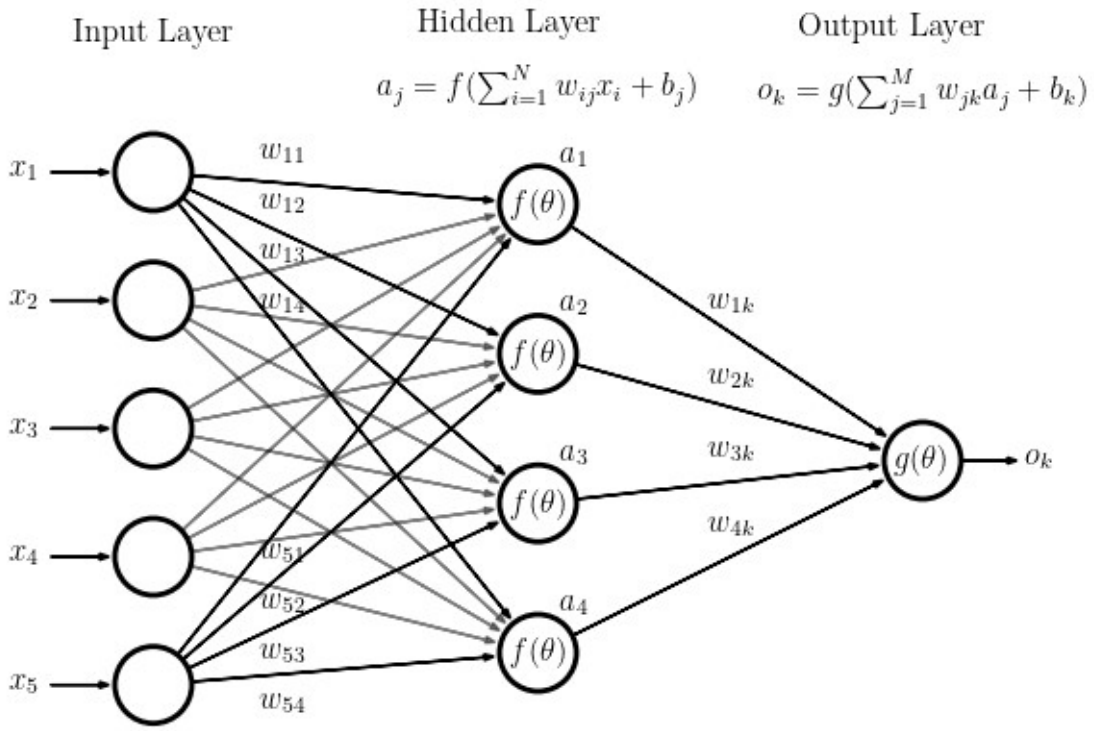


Figure 4.1: Representation of a simple Neural Network with one only hidden layer

The more natural and genuine way in order to model such process is by using an evolutionary algorithm [Câmara, 2015]. Evolutionary algorithms are algorithms which take inspiration from the Darwinian evolutionary theory: they are characterized by the existence of a population of individuals exposed to environmental pressure, which leads to natural selection, i.e. the survival of the fittest, and in turn the increase of the average fitness of the population. Fitness is the measure of the degree of adaptation of an organism to its environment; the bigger the fitness is, the more the organism is fit and adapted to the environment. This kind of algorithms are typically used to solve problems that cannot be easily solved in polynomial time, such as classically NP-Hard problems, and anything else that would take far too long to exhaustively process [Soni, 2018].
From an high level point of view, in our designed evolutionary scenario the agents take decisions, and, given their outcomes, they will reproduce unequally according on how they perform, by following some preset selection rules. The population parameters will evolve (mutate) according to predefined rules. The intelligence or ability to generalized is intended as memory: the agents will initially behave randomly but they will each time evaluate their choices and outcomes, remembering when they failed and succeed and learning from their experience.
Evolutionary algorithms are already successfully used in the process of neural networks training. As an easy implementation example, I report [Fogel et al., 1990], one of the first paper reviewing evolutionary programming as a technique for training neural networks; in this paper, the author claimed that an evolutionary approach can yield faster, more efficient and robust training procedures.
The evolutionary technique is based on the premise of natural selection, and taking inspi-

ration from the example reported in [Lazarou, 2019], it can be seen as a five-step process:

1. Create an initial population of organisms, in our case neural networks.
2. Evaluate each organism based on some criteria, such as the fitness score.
3. Make the agents reproduce according to a selection rule.
4. Mutate the offspring.
5. Take the new mutated offspring population and return to step two. Repeat until some conditions are met (e.g. a fixed number of generations pass, a target fitness is achieved, etc.)

What happens inside each of these steps is deepen in the Chapter 5.
In this work we will use the algorithm only to evolve the agents' weighting strategy to take decisions, i.e. just modifying weights and biases of the network.
Generally speaking, an evolutionary training algorithm can be used to evolve all the network parameters, optimizing the weights, the learning rules and/or the entire network architecture. Therefore, for completeness, I would like to spend some words on the promising application of evolving neural networks in their integrity and on its outstanding advantages. The architecture of a network, i.e. how the neurons are connected to each other, plays a very important role in whether or not an Artificial Neural Network can be trained to successfully learn a task. Over the years, experts of the field carefully designed complex architectures to achieve and often surpass human-level performance on many different tasks. However, there is no single architecture that fix well for all tasks, and therefore designing performing Artificial Neural Network is a very challenging, time consuming and complex work, requiring knowledge and experience [Harvey, 2017]. In order to choose the best network, many architecture should be evaluated: a ten layer network, for example, can have $\sim 10^{10}$ candidate networks. This makes it impractical to evaluate all combinations and select the most performing one. Finding the best architecture requires answering many questions, such as how to design the components of the architecture, how to put them together, and how to set the parameters [Frolov, 2018] [Real, 2019]. To overcome the weaknesses of the current training algorithms, a lot of effort is currently put into using alternative automated methods to find a good architecture to train neural networks. These algorithms are the evolutionary algorithm described above, which are inspired by natural selection and genetic evolution, attempting to mimic natural process of genetic mutations, crossover and selection, while trying to solve objective function optimization problem. As claimed by the Uber AI Labs team, the process of using evolutionary techniques is "yet another example that old algorithms combined with modern amounts of computing can work surprisingly well" [Stanley and Clune, 2017].

4.4 Evaluate the agents: Reverse Engineering

Our final goal is to understand how evolution in a complex scenario affected our decision making processes. In order to asses this, we train agents both in different complex scenarios (uncertain probabilities) and in a simplified scenario (fixed probabilities). After the training, in order to compare their behaviour, we analyse them making face some choices

from the same out-of-sample dataset, with fixed probabilities.
In order to discover which theory among the ones described in Chapter 2 will best fit them, we develop a "Reverse Engineering Toolbox", allowing as to map the strategy of the evolutionary selected NNs to decision making theories assumptions.
The result we expect is that agents trained in a simplified scenario will act "more rationally", i.e. using an approach similar to an expected utility framework. Conversely, we expect that the agents trained in a complex scenario will be better described by generalized expected utility theories, involving a weighting of objective probabilities.
Following the guidelines of [Ferro et al., 2020], we compare our agents behaviour with two theories, Expected Utility Theory and Rank-Dependant Utility Theory, studying which one best fits our agents' decision making approach.

4.4.1 Methods specifications: EU and RDU

We refer to a generic binary lottery as

$$L = \{x_1^L, p_1^L; x_2^L, p_2^L\} \tag{4.1}$$

which gives outcome x_1^L with probability p_1^L and $x_2^L > x_1^L$ with probability $p_2^L = 1 - p_1^L$, and we consider a choice between two binary lotteries, $L = \{A, B\}$.
Following Expected Utility theory, the utility of each lottery will be:

$$EU(L) = u(x_1^L)p_1^L + u(x_2^L)p_2^L \tag{4.2}$$

Regarding the utility function, which is the same for both Expected Utility and Rank-Dependant Utility theory, we impose the following parameterizations:

$$u(x;\alpha) = \begin{cases} x^\alpha & if \quad \alpha \neq 0 \\ ln(x) & else \end{cases} \tag{4.3}$$

This form is represented in Figure 4.2 for different value of α. We remember that a concave utility function is a sign of risk aversion, while a convex one represents risk seeking. In the case of Rank-Dependant Utility theory, the utility of each lottery will be:

$$RDU(L) = u(x_1^L)w(p_1^L) + u(x_2^L)w(p_2^L) \tag{4.4}$$

where we have introduced the probability weighting, imposing one of the two following parameterizations:

$$w_1(p;\delta) = \frac{p^\delta}{(p^\delta + (1-p)^\delta)^{1/\delta}} \tag{4.5}$$

$$w_2(p;\gamma,\kappa) = exp(-\kappa(-ln(p))^\gamma) \tag{4.6}$$

The probability weighting function $w_1(p;\delta)$ is represented in Figure 4.3, while the probability weighting function $w_2(p;\gamma,\kappa)$ is shown in Figure 4.4; the two parameters κ and γ control the general elevation of the curve and its curvature respectively. In general, the probability weighting function will account for the observed humans tendency to overestimate small probability and underestimate high probability events [Tversky and Kahneman, 1992].

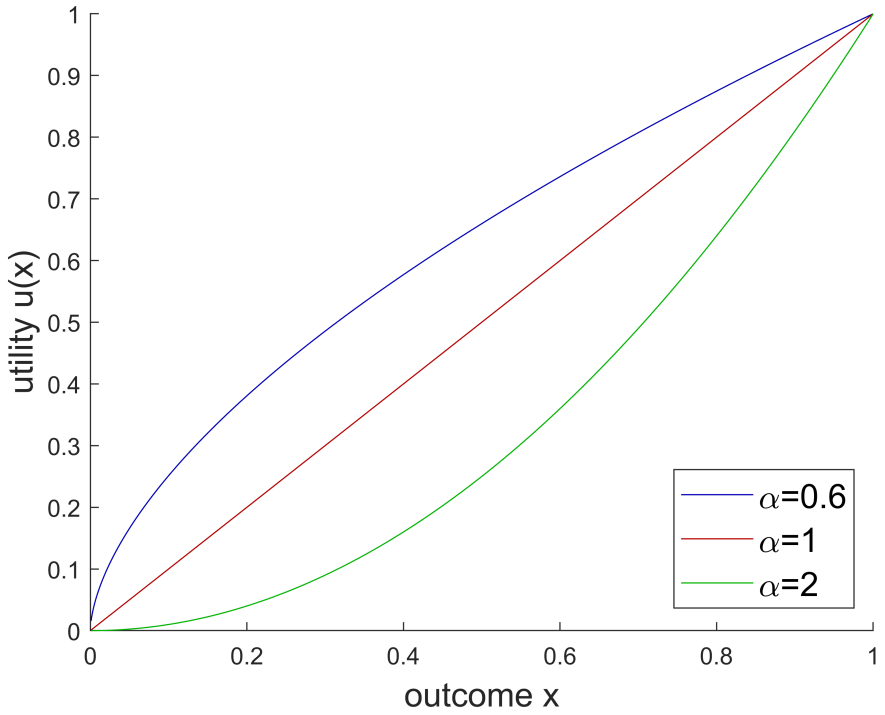


Figure 4.2: Utility function for different values of the parameter α

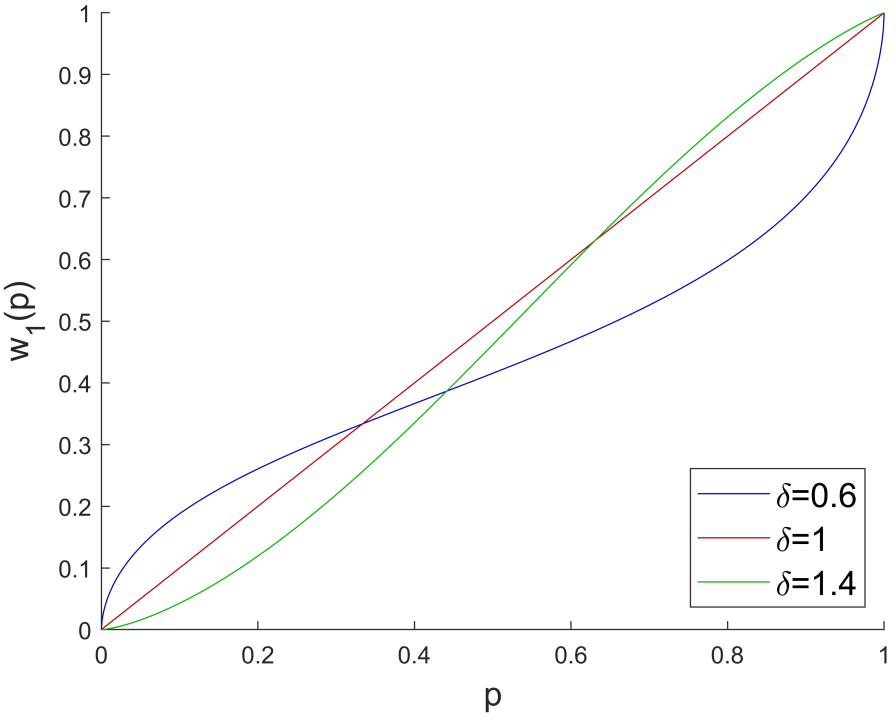


Figure 4.3: Probability weighting function 1st type for different values of δ

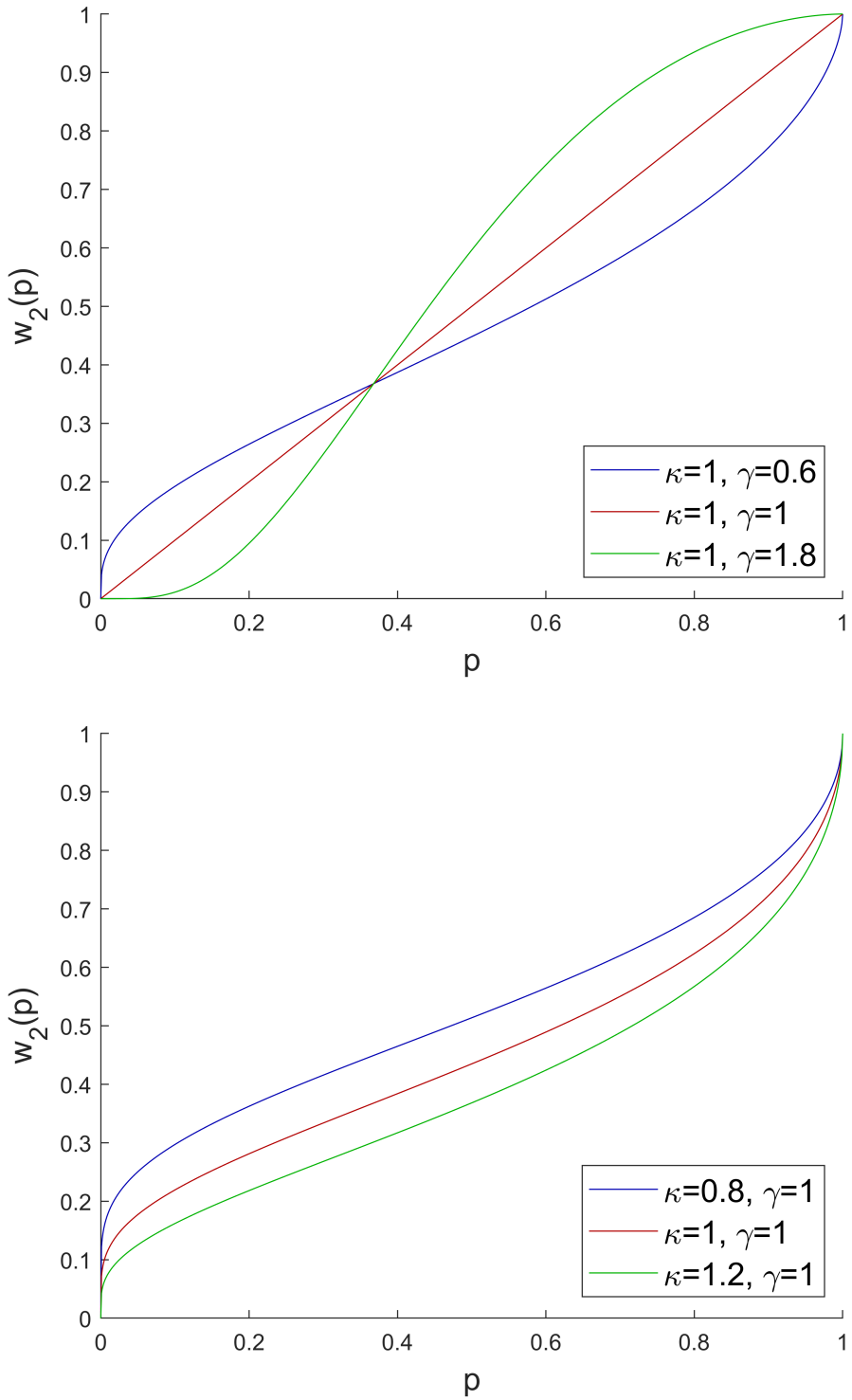


Figure 4.4: Probability weighting function 2nd type for different values of κ and γ

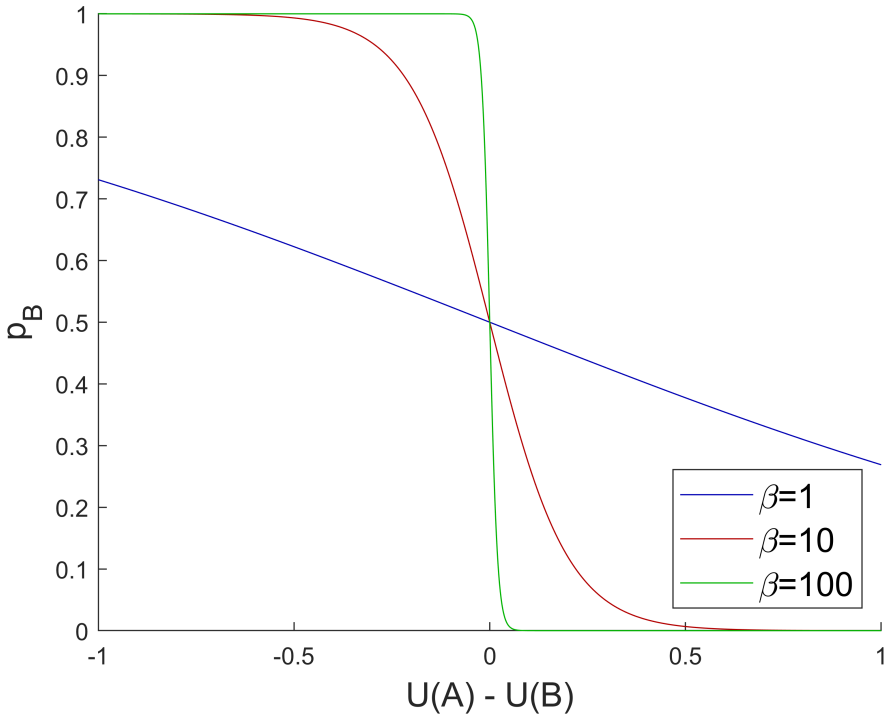


Figure 4.5: Probability of choosing lottery B over lottery A as a function of the difference in utilities

Expected Utility and Rank-Dependent Utility predict that the best option (with the largest utility) will always be selected. In order to account for probabilistic deviations, following the implementation on [Ferro et al., 2020], we introduce a Logistic function, so that the probability of choosing lottery B over lottery A is given by

$$p_B = \frac{1}{1 + e^{\beta(U(A)-U(B))}} \tag{4.7}$$

where, depending on which theory is used, $U(A) = \{EU(A), RDU(A)\}$ and $U(B) = \{EU(B), RDU(B)\}$, evaluated according to Eq. 4.2 or 4.4. Here we introduce β, a parameter describing the sensitivity of the decision maker to the difference in utilities, as illustrated in Figure 4.5. Deterministic choice is recovered by letting $\beta \to \infty$.

4.4.2 Methods evaluation: Maximum Likelihood

All the analyzed models are probabilistic: thus, we can easily implement a maximum likelihood estimation (MLE), as criterion to choose which theory best fit the agents behaviour. After the training phase, the answers of an agents i will be computed as output of the Neural Networks and stored in an array of length L. The total evaluated output will be then stored in a matrix S of dimensions NxL, of which the matrix element $s_{i,j}$ will be the i-th agent choice in the j-th lottery, equal to an ising-like variable:

$$s_{i,j} = \begin{cases} -1 \text{ , if subject i chooses A in the gamble j} \\ +1 \text{ , if subject i chooses B in the gamble j} \end{cases} \tag{4.8}$$

Performing the simple transformation

$$\Phi_{i,j} = \frac{s_{i,j} + 1}{2} \tag{4.9}$$

we encode the output information in $\Phi_{i,j}$, defined as:

$$\Phi_{i,j} = \begin{cases} 0 \text{ , if subject i chooses A in the gamble j} \\ 1 \text{ , if subject i chooses B in the gamble j} \end{cases} \tag{4.10}$$

Then, for each model, we look for the parameters that maximize the following aggregate likelihood function:

$$\mathcal{L}(\vec{m}) = \prod_i \prod_j (p_{Aj}(\vec{m}))^{1-\Phi_{i,j}} (p_{Bj}(\vec{m}))^{\Phi_{i,j}} \tag{4.11}$$

where the productory on i is among the agents, the productory on j is among the choices, and $\vec{m}$ is the parameter vector relative to the model considered (e.g. $\vec{m} = (\alpha, \beta, \kappa, \gamma)$ for RDU_2).
We can simplify eq. 4.13 computing the productory on the agents

$$\mathcal{L}(\vec{m}) = \prod_j (p_{Aj}(\vec{m}))^{N-N_j} (p_{Bj}(\vec{m}))^{N_j} \tag{4.12}$$

where N is as always the number of agents and $N_j = \sum_i \Phi_{i,j}$ the number of agents which at the j-th gamble chose B.
For computational stability, we actually maximize the log-likelihood:

$$log\mathcal{L}(\vec{m}) = \sum_j (N - N_j) p_{Aj}(\vec{m}) + N_j p_{Bj}(\vec{m}) \tag{4.13}$$

$$\vec{m}^* = argmax_{\vec{m}}(log\mathcal{L}(\vec{m})) \tag{4.14}$$

We solved the optimization problem in 4.14 with a MATLAB-routine which implements the Nelder-Mead simplex algorithm, as described in [Lagarias et al., 1998].

4.4.3 Methods comparison: Nested Hypothesis Testing

Once the maximum likelihood estimation is performed for each model, statistical tests are needed to compare the quality of the calibrations of the models to the data.
In order to confront the two models, we can exploit the fact that EU and RDU models are nested, which permits us to employ an efficient comparison strategy. Two models are nested if one model contains all the terms of the other and at least one additional term, i.e. the parameters of one model are a subset of the parameters of the other. The larger model is the complete (or full) model, and the smaller is the reduced (or restricted) model.
When a model R with k_R parameters is nested in a model F with $k_F \geq k_R$ parameters, according to Wilk's Theorem [Wilks, 1938] one can compare the model performances through the likelihood ratio test statistic. Defining the null hypothesis as "the simpler

model R is the true one", it can be shown that, under the null hypothesis, as the sample size n goes to ∞

$$D = 2log\frac{\mathcal{L}_F}{\mathcal{L}_R} \sim \chi^2(k_F{-}k_R) \tag{4.15}$$

where $\mathcal{L}_R$ and $\mathcal{L}_F$ are the maximum-likelihood estimations of the nested model R and of the nesting model F, and $\chi^2(k)$ is a chi-squared distribution with k degrees of freedom. At a significance level ϵ, the null hypothesis is rejected if the p-value p is smaller than ϵ:

$$p = (P(\chi^2(k_F{-}k_R)) \geq D) = 1 - F(\frac{k_F{-}k_R}{2}, \frac{D}{2}) < \epsilon \tag{4.16}$$

where $F(\frac{k_F{-}k_R}{2}, \frac{D}{2})$ is the cumulative distribution function of the chi-square distribution, evaluated in the point D. For a chi-square distribution, $F(s,x) = \frac{\gamma(s,t)}{\Gamma(s,t)}$ is the regularized gamma function and it is the ratio between the lower incomplete gamma function and the ordinary gamma function.
Being the model R the simplest one, in our case R will correspond to EU, F will correspond to RDU and the degrees of freedom of the chi-square k will be $k = k_F{-}k_R = 4 - 2 = 2$ or $k = k_F{-}k_R = 3 - 2 = 1$, depending on which probability weighting function is used.

5. Inside the Agent Based Model

In this section we will go deeper inside the ABM, explaining the meaning of each of its steps.

5.1 Create the initial population

The population of Neural Networks is characterized by the same architecture and each network is initialized with different random weights and biases.
More specifically, regarding the architecture, we fix:

- Input and output dimensions of the organisms
- Output activation function: we choose as output activation function an hyperbolic tangent sigmoid transfer function: $output \in [-1, +1]$, meaning that if the output equals -1 the NN choses the 1st lottery with probability equal to 1, while if the output equals +1 it chooses the 2nd lottery with probability equal to 1
- Complexity of the networks, intended as number of layers and neurons. We designed the agents to be as simple as possible and, after some attempts, we configure the neural networks with one only hidden layer with a number of 10 neurons in it

In order to make sure that the chosen neural networks architecture is good enough to understand and solve the problem of choosing the best lottery among two possibilities, we initially designed a training set classifying "good" and "bad" choices, and we train the neural networks with a classic back-propagation algorithm. In a test phase, it turns out that the chosen architecture was sufficient for the networks to be correctly trained, so we kept the above described architecture.
A diagram representing the used neural network structure is showed in Figure 5.1.

5.2 Define the complex environment

During a lifetime, each agent will face L different lottery choices. This space of lotteries will be extremely complex, with different levels of uncertainty to enable and facilitate the development of heuristics, in a scenario as similar as possible to a real-world one.

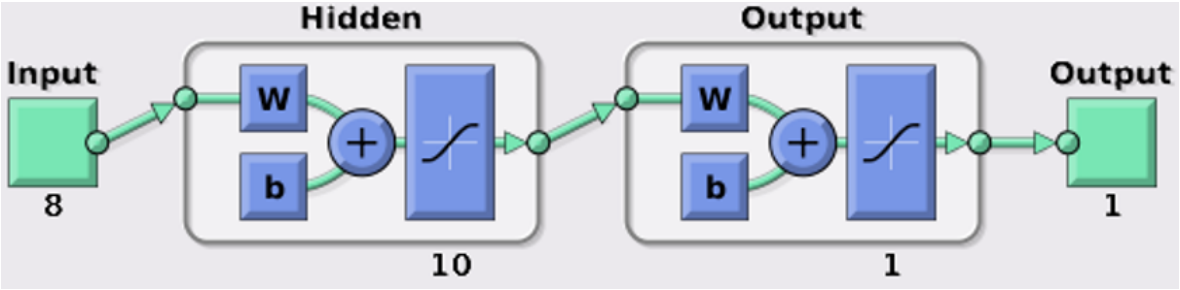


Figure 5.1: Neural Network used as agents in the ABM

5.2.1 Input Dataset: introduction of uncertainty

In this work, for simplicity, we introduce uncertainty only in probabilities, while the payoffs are fixed. Being the developed model extremely tunable, a possible extension can easily include uncertain payoffs as well.
We build the input dataset in the following way: each agent, represented by a Neural Network, during his life (one time step of the model) will face L subsequent binary choices, each of two alternatives being a binary lottery. Hence, the input structure of the NN will be a binary choice between binary lotteries, both selected from an ensemble of lotteries.
We now focus on a single binary lottery to characterize the structure. If the lottery A have had known probabilities, we would construct it in the following way:

- Choose two possible outcomes from the set $X -> x_i^A, x_j^A$
- Choose a probability of outcome $x_i^A -> p_i^A$
- The lottery is then: $A = (x_i^A, p_i^A; x_j^A, 1 - p_i^A)$

However, in the training phase, we want the agents to face uncertain setups. The simplest generalization involves (partially) unknown event probabilities:

- Choose two possible outcomes from the set $X -> x_i^A, x_j^A$
- Choose an interval of possible probabilities for the outcome x_i^A, such that the unknown probability p_i^A of outcome x_i^A, lies in $[p_i^{A,min}, p_i^{A,max}]$
- The lottery is then:

$$A = (x_i^A, [p_i^{A,min}, p_i^{A,max}]; x_j^A, 1 - [p_i^{A,min}, p_i^{A,max}]) \tag{5.1}$$

Each binary lottery has thus 4 parameters: $(x_i^A, x_j^A, p_i^{A,min}, p_i^{A,max})$.
A training input is a decision task l between two lotteries A_l and B_l, being characterized by 8 parameters (4+4):

$$l = (x_i^A{}_l, x_j^A{}_l, p_i^{A,min}, p_i^{A,max}, x_i^B{}_l, x_j^B{}_l, p_i^{B,min}, p_i^{B,max}) \tag{5.2}$$

At each decision task, agents make a choice between the two lotteries. Then the outcome of the chosen lottery is extracted, i.e. for one single lottery the probability p_i is uniformly sampled between p_i^{min} and p_i^{max}. Finally the payoff is computed for each agent and the evolutionary algorithm proceeds as described later on the Chapter.

5.2.2 Uncertainty targeting: different scenarios

Considering again one single lottery, we now will make some assumptions in order to characterize the problem. First of all, we choose the two outcomes of the lottery (x_i, x_j) to be in the interval $[0, 1]$. More precisely x_i is extracted following different distributions, while x_j is taken fixed and equal to 1: $x_i \leq x_j = 1$. This measure was taken in order to have an easier characterization of the level of uncertainty of the problem: depending on the distribution of the variable x_i and on its probability distribution, we will have different levels of uncertainty.
It is important to stress that the concept "uncertainty" is quite complex to define; it can be interpreted as level of lack of probability knowledge (for example as the length of the interval $p_{max} - p_{min}$ in which the actual probability can fall) or it can be interpreted as uncertainty of the final result, including a combination of both probabilities and related outcomes. Following the second interpretation, and taking fixed the second outcome $x_2 = 1$, we characterized four possible scenarios:

1. **WU**, or Weak-Uncertain scenario.
   In this scenario x_1 is large ($x_1 \sim 1$) and it is extracted with low probability ($p_1 \sim 0$). Therefore, all the possible outcomes (x_1 and x_2) are large and the agents are subject to weak uncertainty, in the sense that whatever the outcome will be, they will be well rewarded.

2. **U1**, or Uncertain-1st-Type scenario.
   In this scenario, x_1 is large ($x_1 \sim 1$) and it is extracted with high probability ($p_1 \sim 1$). Therefore, agents will obtain with high probability the lower outcome x_1; however, the value of x_1 will still be high, hence this type of scenario will be characterized by a low degree of uncertainty.

3. **U2** or Uncertain-2nd-Type scenario.
   In this scenario, x_1 is small ($x_1 \sim 0$) and it is extracted with low probability ($p_1 \sim 0$). Therefore, agents will obtain with high probability the higher outcome x_2; however, if x_1 will be extracted, the agents will receive a small outcome, hence this scenario is described by an higher uncertainty, compared to the previous ones.

4. **SU** or Super-Uncertain scenario.
   In this scenario, x_1 is small ($x_1 \sim 0$) and it is extracted with high probability ($p_1 \sim 1$). Therefore, agents will obtain with high probability the lower outcome x_1, hence this scenario is described by super uncertainty.

In order to characterize these worlds, we sampled the value of x_1 and of the two probabilities p_{min} and p_{max} from two different beta distributions: $B(\alpha_1, \beta_1)$, which is peaked around 0, and $B(\alpha_2, \beta_2)$ which is peaked around 1. The two beta distribution are represented in Figure 5.2 and the values used for the simulations are $\alpha_1 = 0.5, \beta_1 = 3, \alpha_2 = 3, \beta_2 = 0.5$. The table 5.1 schematize the procedure used to create the different scenarios, where $\vec{p} = \{p_{min}, p_{max}\}$.

	WU	U1	U2	SU
x_1	big: $x_1 \sim B(\alpha_2, \beta_2)$	big: $x_1 \sim B(\alpha_2, \beta_2)$	small: $x_1 \sim B(\alpha_1, \beta_1)$	small: $x_1 \sim B(\alpha_1, \beta_1)$
$\vec{p}$	small: $\vec{p} \sim B(\alpha_1, \beta_1)$	big: $\vec{p} \sim B(\alpha_2, \beta_2)$	small: $\vec{p} \sim B(\alpha_1, \beta_1)$	big: $\vec{p} \sim B(\alpha_2, \beta_2)$

Table 5.1: Different world scenarios creation

In addition to this four world-scenarios, we developed other two worlds with more extreme conditions:

5. **FL** or Floating scenario.
   Here x_1 is extracted from a less peaked beta distribution $B(\alpha_3, \beta_3)$, with parameters $\alpha_3 = 0.5$ and $\beta_3 = 1$, while p_{max} and p_{min} are extracted from a beta distribution $B(\alpha_4, \beta_4)$ with parameters $\alpha_4 = 0.5$ and $\beta_4 = 0.5$; the two distribution for x_1 and for $\vec{p} = \{p_{min}, p_{max}\}$ are shown in Figure 5.3. This world is developed to train the agents in a super-floating scenario; indeed, three different situations can happen:

   - both p_{max} and p_{min} are small: p will be small
   - both p_{max} and p_{min} are big: p will be big
   - p_{max} is big and p_{min} is small: p will be very uncertain

6. **CE** or Certain scenario.
   In this world the probability are fixed, hence $p = p_{max} = p_{min}$, extracted from an uniform distribution, while x_1 is extracted from the same distribution as for the Random world ($B(\alpha_3, \beta_3)$ in Figure 5.3).

5.3 Make the agents learn: Evolutionary Algorithm

5.3.1 Evaluate the fitness

Measuring how well an organism perform is the crux of evolutionary algorithm design. In general, fitness functions are task-specific, i.e. "good performance" varies from task to task; for example, in a classification problem, a way to evaluate a performance may be classification accuracy.
To measure how well an organism performs, since in a really uncertain environment is complex to determine a criterion to assess the optimal choice, we evaluate the agents' fitness from confronting one agent's performance with the rest of population performances. There are substantially two ways for comparing an agents' performance with others', i.e. following a:

- Cardinal utility, i.e. considering how much each agent actually gained and comparing the values.
- Ordinal utility, i.e. considering the ranking among agents' outcomes.

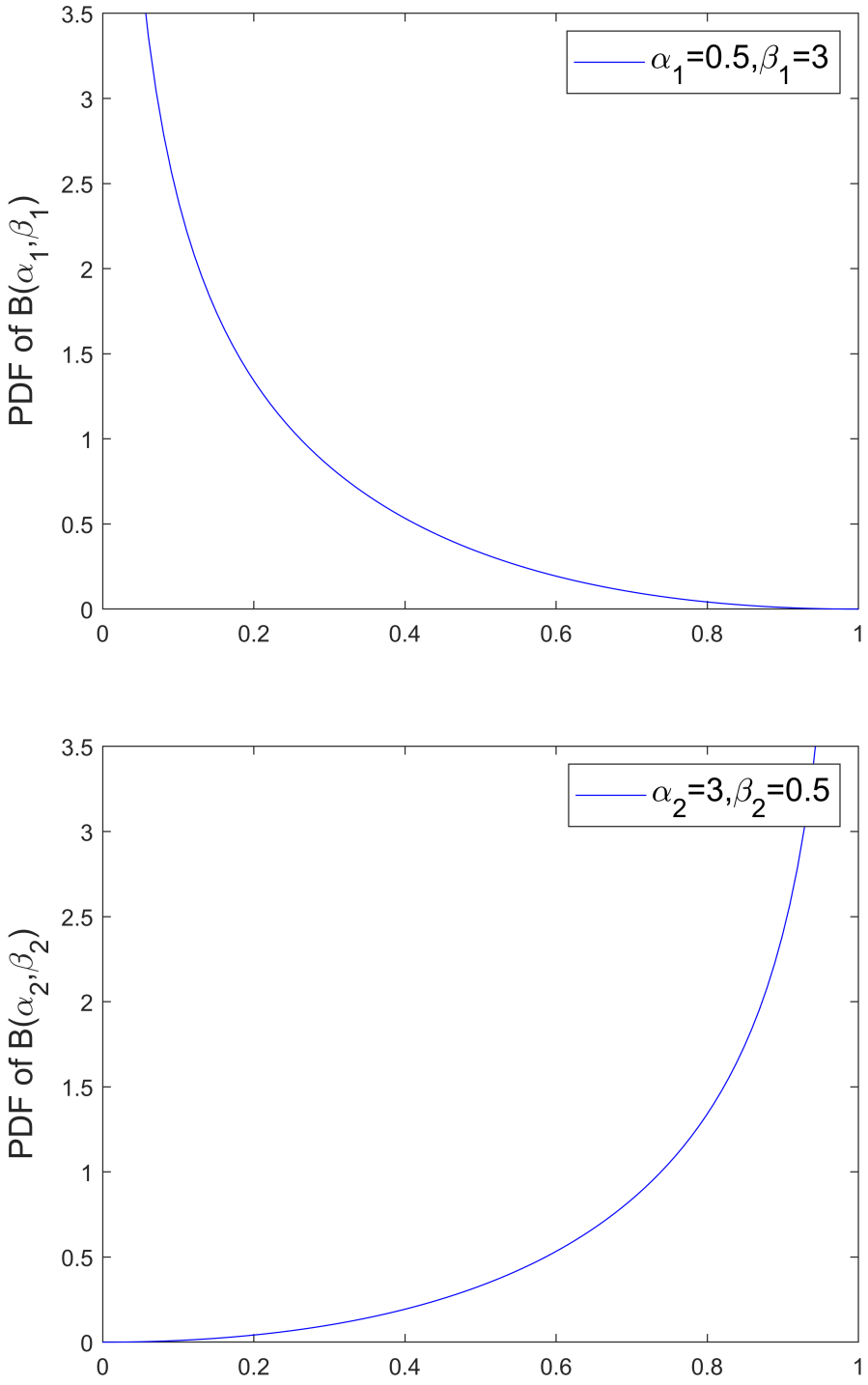


Figure 5.2: The two probability distribution functions $B(\alpha_1, \beta_1)$ and $B(\alpha_2, \beta_2)$ describing the worlds WU, U1, U2 and WU.

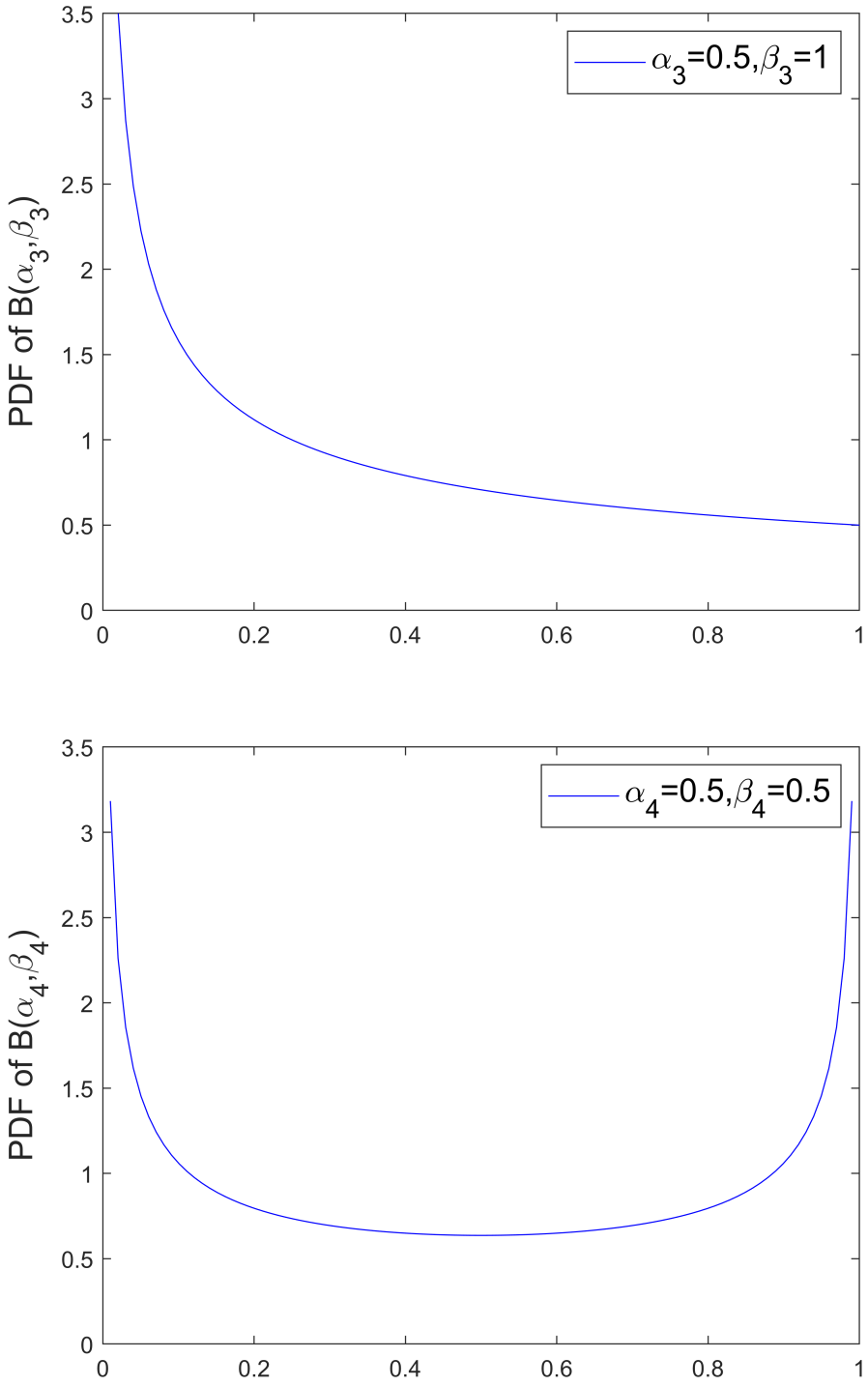


Figure 5.3: The two probability distribution functions $B(\alpha_3, \beta_3)$ and $B(\alpha_4, \beta_4)$ describing the Floating scenario. The first picture represent the probability distribution from which x_1 is extracted, the second the one from which p_{min} and p_{max} is extracted.

In this work, we will analyse both approaches: cardinal ranking with proportional selection and ordinal ranking with truncation selection and sigmoid-weighted selection.
In order to compute the fitness, the first step is to compute the total payoff of each agent, intended as the sum of the single lotteries payoffs. If the agent i faces in his lifetime L choices, his total payoff will be:

$$payoff_i = \sum_{j=1}^{L} S_{i,j} \tag{5.3}$$

where $S_{i,j}$ is the outcome gained by the i-th agent in the j-th choice. After having computed the payoff, we developed three above described different way of computing the fitness; in any of the following methods the fitness is a number included in the interval $[0, 1]$.

5.3.1.1 Proportional Selection

Using this algorithm, we simply define the fitness through a normalization of the payoffs. For the i-th agent:

$$fitness_i = \frac{payoff_i}{\sum_{i=1}^{N} payoff_i} \tag{5.4}$$

where N is the total number of agents.
Although proportional selection could seem the simplest and more natural method, it assumes too many linear proportionalities. This method is indeed based on the concept of cardinal utility, where preference orderings are preserved uniquely up to increasing monotone linear transformations. This hypothesis is sometimes too strong and ordinal utility, with its weaker assumptions (preference orderings preserved up to increasing monotone transformations), is usually preferred in consumer choice theory because it best represents reality.
According to ordinal utility, it is only meaningful to ask which option is better, but it is meaningless to ask how much better. We therefore figured two other different ways to compute the fitness based on ordinal ranking of payoffs, i.e. mapping the possible ranked payoffs to a fitness function. These two methods are truncation selection and sigmoid-weighted selection (which is based on ordinal ranking but also resumes some aspect of cardinal utility).

5.3.1.2 Truncation Selection

In this method, the function mapping the payoffs to the corresponding fitness is a step function. This selection rule is really simple: a certain percentage $\%P$ of the population, which we want to survive and transfer the genes, is chosen. Then the payoffs are sorted and the $\%P$ agents with highest payoffs will reproduce with equal probability, i.e. we assign to the surviving agents the same positive fitness, while we assign to the non-surviving ones a zero fitness. Obviously, if the $\%P$ of the agents is not an integer number, we will round it to the next integer value. The number of surviving agents therefore is $\lceil \%P \cdot N \rfloor$, where N is as always the number of agents.

$$fitness_i = \begin{cases} \frac{1}{\lceil \%P \cdot N \rfloor} & if \quad payoff_i \in \lceil \%P \cdot N \rfloor \quad best \quad payoffs \\ 0 \quad else \end{cases} \tag{5.5}$$

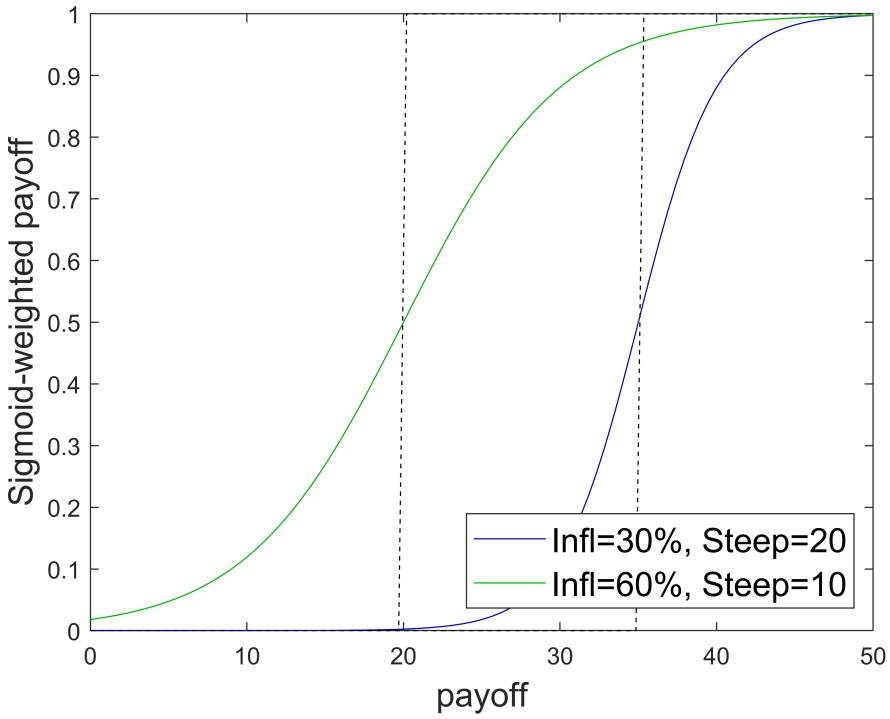


Figure 5.4: Mapping payoff-fitness under sigmoid-weighted selection for two different couple of values steepness ξ and inflection point β

5.3.1.3 Sigmoid-weighted Selection

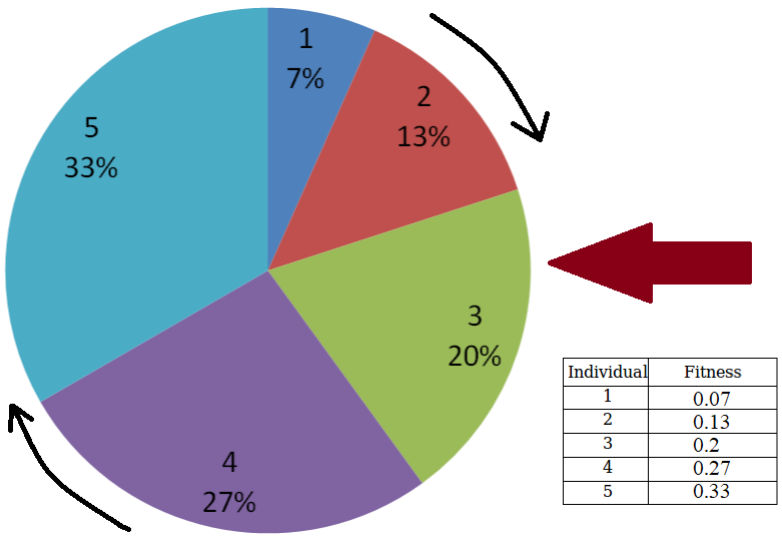
This reproduction scheme allows a gradual departure from the proportional selection scheme towards a fitness function that is more similar to the one created by truncation selection. It is also based on an ordinal ranking, and for these reason we can interpret this method as a softer version of truncation selection: instead of mapping fitness to payoffs following a step function, it uses a sigmoid function. Differently from truncation selection, here the fitness, and so the probability of reproduction, of the survivors will have different values. The number of offspring probabilistically assigned to each individual is proportional to its payoff after the following transformation:

$$weighted\_payoff_i = \frac{1}{1 + e^{-\xi(p_{N,i} - \beta)}} \tag{5.6}$$

where ξ represents the steepness of the sigmoid, $p_{N,i}$ the individual's sum of payoffs after normalization, such that the highest-scoring individual has a sum of payoffs equal to 1, and β represents the sigmoid's inflection point and it is correlated to the percentage of surviving agents. After the weighting, the same procedure of proportional selection is applied:

$$fitness_i = \frac{weighted\_payoff_i}{\sum_{i=1}^{N} weighted\_payoff_i} \tag{5.7}$$

This mapping payoff-fitness is taken from the paper [Kolodny and Stern, 2017] and two examples are represented in Figure 5.4.



Individual	Fitness
1	0.07
2	0.13
3	0.2
4	0.27
5	0.33

Figure 5.5: Representation of wheel selection rule

5.3.2 Agents reproduction

The reproduction step has itself two steps: parent selection and progeny creation. Each new organism needs k parents: in asexual reproduction k is 1, while in sexual reproduction k is 2 or more. Deciding which organisms will parent each child should be done on the basis of their fitness score, where the fittest organisms should produce more offspring. In this work we used for simplicity k=1, i.e. asexual reproduction; the son will be an exact copy of the parent, inheriting the weights and biases of the NN (only in the subsequent step will be added a mutation).
We impose non-overlapping generations, hence we replace all the agents.
An individual can become a parent with a probability which is proportional to its fitness. Therefore, fitter individuals have a higher chance of reproduction and of propagating their features to the next generation. This applies a selection pressure to the more fit individuals in the population, evolving better individuals over time. Specifically, we apply the roulette wheel selection rule, called in this way because it could be imagined similar to a Roulette wheel in a casino. In this method, a proportion of the wheel is assigned to each of the individual based on their fitness value; then a random selection is made similarly to how the roulette wheel is rotated. A fitter individual has a greater pie on the wheel and therefore a greater chance of being selected. In the simplest case of truncation selection, each of the surviving individuals will have the same portion of the wheel, meaning that, for each replacement, a parent will be extracted uniformly among all possible parents. The wheel selection rule is represented in Figure 5.5.

5.3.3 Agents mutation

Mutation is a fundamental step in any evolutionary algorithm, since it permits to explore more states, accelerating the learning process to quickly reach an evolutionary stable state [Mallipeddi et al., 2011].
In this work, the mutation step is realized as the addition of Gaussian noise to each weight

and bias in the network. We do not change the activations or architecture of the network, although a more advanced evolutionary algorithm could certainly do so, by adding or removing nodes in the hidden layers.
The gaussian noise is proportional to "how bad" the agents' perform, and a mutation is applied in two different timescale of the algorithm, hence we can divide in two class of learning:

- Intra-generational: in between the agents lifetime
- Inter-generational: in between different generations

Generally speaking, the intra-generational mutation will be smaller with respect to the inter-generational one, reflecting the fact that between generations, when agents are replaced, bigger changes should be observed.

5.3.3.1 Intra-generational mutation (learning)

Any agent during his lifetime faces L choices. After having played the j-th lottery at time t, we compute the mean over the payoff-outcomes of the j-th lottery (the so called cost of the j-th lottery) as:

$$cost_j(t) = \frac{1}{N}\sum_{i=1}^{N} S_{i,j}(t) \tag{5.8}$$

where $S_{i,j}$ is the outcome gained by the i-th agent for the j-th choice.
A mutation is then added whenever the lottery payoff of the single agent is lower than the cost, i.e. lower then the mean outcome for the single lottery. The mutation, added to all the weights and biases of the network, is an array of Gaussian random variables with mean 0 and variance proportional to how bad an agent performed in that single choice. Calling wb_i the vector of weights and biases of the network, and supposing that the choice is taken at time t, the vector of weights and biases for the agent i after one single choice at time $t + \delta t$ will be:

$$wb_i(t+\delta t) = \begin{cases} wb_i(t) & if \quad S_{i,j}(t) > cost_j(t) \\ wb_i(t) + \xi_i(t) & else \end{cases} \tag{5.9}$$

where

$$\xi_i(t) \sim N(0, \epsilon \cdot (cost_j(t) - S_{i,j}(t))) \tag{5.10}$$

and ϵ is a small number to make the variance of the Gaussian a reasonable number.

5.3.3.2 Inter-generational mutation

The offspring, as already explained, is initially created as an exact copy of the selected parent. In the subsequent moment, a mutation is applied to the whole offspring, proportional to how much the parent misperform during his lifetime compared to other agents.
Also in this case, the mutation is added to all the weights and biases of the network, as an array of Gaussian random variables with mean 0 and variance proportional to how bad an

agent performed.
The offspring having as parent the i-th agent will be:

$$wb_i = wb_i + \zeta_i \tag{5.11}$$

where

$$\zeta_i \sim N(0, \kappa \cdot (payoff\_max - payoff_i) \tag{5.12}$$

where $payoff\_max$ is the maximum payoff among the total payoffs of all agent and κ is a small number to make the variance of the Gaussian a reasonable number.
We suppose the inter-generational mutation bigger than the intra-generational mutation, hence $\xi_i < \zeta_i$.

5.3.4 Repeat across generations

It is well known that very basic rules repeated continuously can give rise to arbitrary complexity: basically, one can form a complex network just by applying simple rules and iterating many times, and this is one of the fundamental feature of our agent based model. These previous steps characterized the process of one only generation, which is defined as the accomplishment of the entire set of L choices. When the population have taken all the decisions, they finish their lifetime and they are all replaced: for simplicity, as already mentioned, we choose the framework of non overlapping generations. As soon as the old agents are replaced with new agents, the offspring, the new agent are faced to the L subsequent choice and the algorithm restarts. We make the agents evolve until a stationary state is reached or a fixed number of generations have passed.

5.4 Out-of-sample analysis

While in the training part agents develop heuristics in complex scenarios, in the out-of-sample analysis part, agents apply these short-cuts to new situations. The purpose of this final step of the algorithm is to evaluate and understand our agents behaviour.
After having built an out-of-sample dataset to be faced to the trained agents, one can either evaluate the learning process and agents performance by means of scoring rules or mapping the agents strategy to some existing decision making theories (with the reverse engineering procedure, explained in 4.4).

5.4.1 Out-of-sample dataset

After the training phase, comes the out-of-sample analysis, where the evolutionary selected agents face different choices with respect to the ones faced during the training. We define the set of choices faced by the agents in this final phase as out-of-sample dataset, suggesting the fact that this dataset can be substantially different from the one tackled in the training phase.
Specifically, the out-of-sample can be of the same form of the training dataset (hence characterized by the same level of uncertainty) or can include much simpler decision tasks, involving lotteries with known probability. We analyse the agents' performance with the following different typology of out-of-samples:

1. With the training set
2. With an out-of-sample built in the same way of the training set, i.e. with uncertain probabilities, representing real-world-complexity
3. With an out-of-sample built in a different way with respect to the training set, i.e. with certain probabilities. This represents the oversimplified scenario of laboratory-like choices. A typical input belonging has 8 parameters, of which therefore only 6 are non degenerate, i.e. $p^{min} = p^{max}$, and therefore it is a particular case (just in the form) of the input used in the training phase.

5.4.2 Different scoring rules to evaluate performances

To evaluate the agents' performance, we take an out-of-sample dataset and pursue the following steps:

1. Compute the so called rational-choices for the out-of-sample dataset, based on higher expected value of the lotteries. The vector of rational-choices will be a vector of length L, where L is the number of choices between a first lottery A and a second lottery B, of which the element r_j will be:

$$r_j = \begin{cases} -1 & if \quad EV(A) \geq EV(B) \\ +1 & else \end{cases} \tag{5.13}$$

   where EV(A) (resp. EV(B) is the expected value of the lottery A (resp. B).

2. Deciding the metrics of interest: Having N agents and wanting to evaluate a single output per system, we designed two different metrics in order to compute a final output:

   - Mean Output:
     We first compute the output of each NN, simulating the Neural Networks on the out-of-sample set and evaluate its choices: Collecting all the outputs in a matrix M of dimension NxL, each matix element $m_{i,j}$ will be in the interval $[-1, 1]$. Giving this number we have to evaluate the choices, that we will collect in a matrix S, always of dimension NxL. Considering negative output as choosing the first lottery and second output as choosing the second lottery, the matrix element $s_{i,j}$ will be the choice of the i-th agent in the j-th lottery:

$$s_{i,j} = sgn(m_{i,j}) = \begin{cases} -1 \text{ , if subject i chooses A in the choice j} \\ +1 \text{ , if subject i chooses B in the choice j} \end{cases} \tag{5.14}$$

     where sgn is the sign function.
     We then compute the mean output for each of the L choices, averaging between all the N agents. For the j-th lottery, the final output will be:

$$o_j = \frac{1}{N} \sum_{i=1}^{N} s_{i,j} \tag{5.15}$$

- Representative agent's output:
  We design a representative agent; this agent has as vector of weights and biases wb_{rep} the mean of all agents' weights and biases.

$$wb_{rep} = \frac{1}{N}\sum_{i=1}^{N} wb_i \tag{5.16}$$

  Then we will compute the outputs $m_j \in [-1, 1]$ of the representative agent for the L choices. For the j-th lottery, the final output will be:

$$o_j = sgn(m_j) \tag{5.17}$$

where sgn is always the sign function.

3. Evaluate the performances, according to different scoring rules:

   - Accuracy, defined as measure of closeness of the outputs. Computed by counting the fraction of the agent's choices equal to rational choices:

$$Acc = \frac{1}{L}\sum_{i=1}^{L} f_j(r_j, o_j) \tag{5.18}$$

     where

$$f_j(r_j, o_j) = \begin{cases} 1 & if \quad r_j = o_j \\ 0 & else \end{cases} \tag{5.19}$$

   - Mean square error (or Brier score), defined as measure of the difference between the values of rational choice and agent's choice:

$$MSE = \frac{1}{L}\sum_{i=1}^{L}(o_j - r_j)^2 \tag{5.20}$$

   - Final wealth, defined as total amount of payoff acquired in L lotteries. We play all the lotteries and, according on the previous output choice o_j, we compute the total payoff

5.4.3 Study the agents behaviour: Reverse Engineering

After having built a specific out-of-sample dataset, we face our agents with these choices and collect their decisions. We then map their outputs to the models of both Expected Utility Theory and Rank Dependent Utility Theory, finding which is the decision making model that best fit the agents' output, as it is described in Section 4.4. We finally observe and make consideration on the resulting curves of utility function for EU and probability weighting for RDU.

5.5 Summary of the ABM

5.5.1 List of parameters of the ABM

In this subsection we will list all the parameters of the ABM; the values in the parentheses represent the values chosen and used as default, unless otherwise specified.
The most important parameters of the agent based model involve:

- **N**: Number of agents (N=100)
- **Ngen**: Number of generations (Ngen=500)
- **L**: Number of choices per generation, i.e. length of training dataset (L=150)
- **Training scenario**: type of training dataset scenario among WU, U1, U2, SU, Floating and Certain
- **Selection rule**: type off selection rule among proportional, truncation and sigmoid-weighted selection

Then we have several secondary parameters, as:

- Number of choice in the out-of-sample set (50)
- Amplitude of iter-generational mutation (0.005)
- Amplitude of intra-generational mutation (0.001)
- Truncation percentage (for truncation and sigmoid-weighted selection)
- Steepness of sigmoid for sigmoid-weighted selection (10)

And finally the parameters describing the Neural Networks:

- Number of hidden layers (1)
- Number of neurons in the hidden layers (10)
- Hidden layer transfer function (hyperbolic tangent sigmoid)
- Output transfer function (hyperbolic tangent sigmoid)

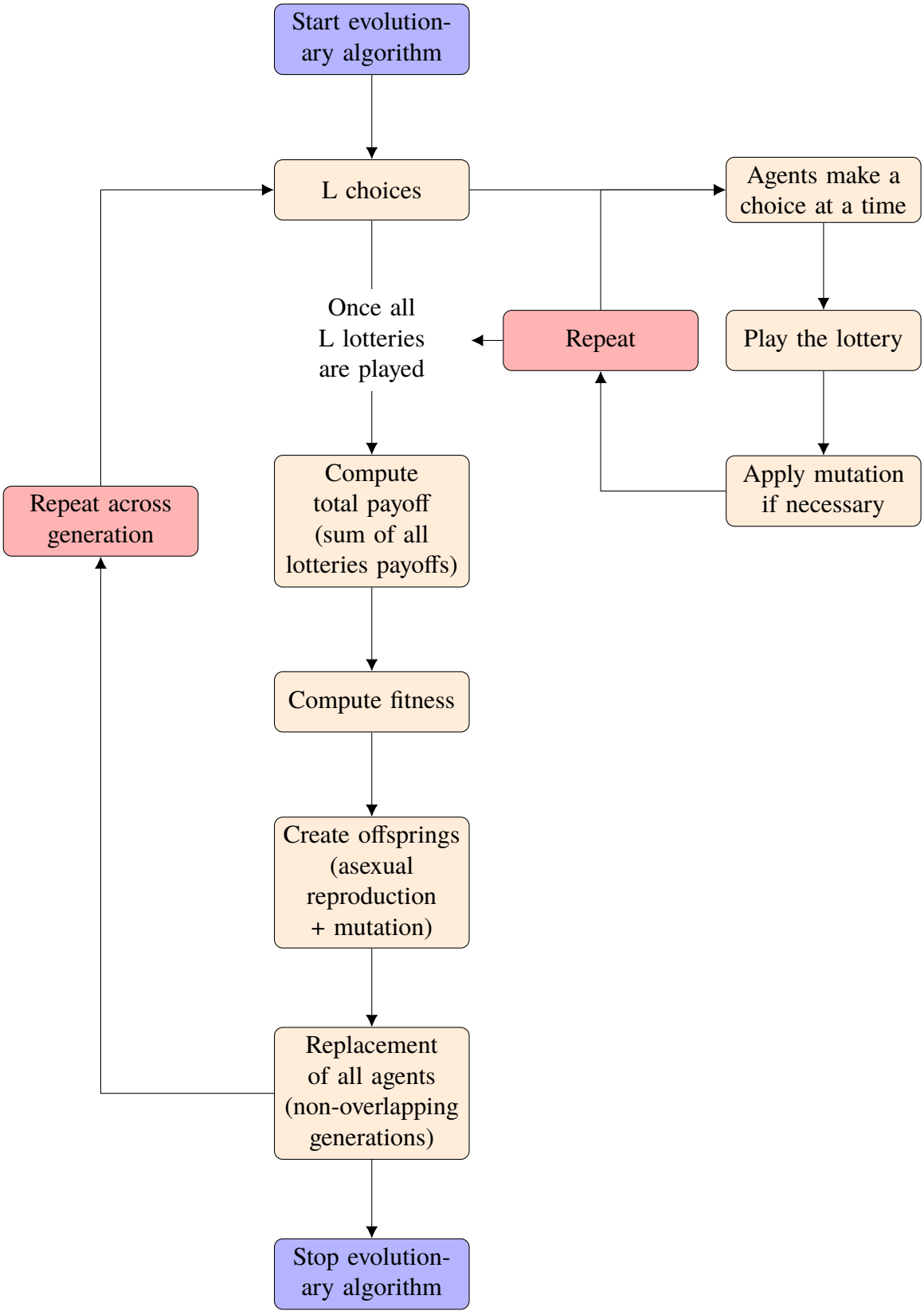
Counting them, we have in total 14 different parameters. Manage this huge number of parameters in the simulations was one of the biggest challenge of this master thesis.
For running the simulations, we use MATLAB as programming language.

5.5.2 Flowchart

To summarize as much as possible, we can regroup the whole algorithm in 4 main steps:

1. **Agents creation**: initialization of NNs population
2. **Environment characterization**: creation of datasets collecting choices between binary lotteries
3. **Agents learning**: training the NNs by means of an evolutionary algorithm
4. **Agents evaluation**: out-of-sample analysis

The third point includes a lot of processes, which are repeated for as many times as the chosen number of generations. To clarifying the ideas and simplify the concept, I present a flowchart showing the cycle of the evolutionary algorithm, to be repeated among the Ngen.


Start evolution-
ary algorithm
L choices
Agents make a
choice at a time
Once all
L lotteries
are played
Repeat
Play the lottery
Compute
total payoff
(sum of all
lotteries payoffs)
Apply mutation
if necessary
Repeat across
generation
Compute fitness
Create offsprings
(asexual
reproduction
+ mutation)
Replacement
of all agents
(non-overlapping
generations)
Stop evolution-
ary algorithm

6. Results

In this section we will present the results of our Agent Based Model.
We can divide the analysis in two main parts: evaluation of the learning process by means of a scoring rule and mapping the agents strategy to existing decision making theories.

6.1 Assess the algorithm: Accuracy Computation

As described in Section 5.4, we defined three different score functions in order to evaluate the agents performances: accuracy, mean square error and final wealth. Each of this scoring rule led to similar results and corresponding conclusions, hence in this chapter we will present only the accuracy-related results.
In this section we will perform a first assessment of the functioning of the evolutionary training algorithm of the Agent Based Model. The question that we address is then: "are our agents actually learning a reasonable strategy to make choices?". In order to answer this question we compute the accuracy, as explained in Chapter 5.4, at the end of each generation-step, for different ABM parameters settings.
Investigating the accuracy as a function of number of generations, we are able to monitor the learning of the agents, understand if they reach an equilibrium state and how long it takes, depending on the different conditions in which the system was set. Moreover, looking into the oscillation of the accuracy, one can outline the different learning behaviour and make considerations with the aim of finding the ABM features and parameters that best mimic reality.
We compute the accuracy in function of generations both on the training set and on an out-of-sample dataset built in the same way of the training set (same type of world-scenario described in Subsection 5.2.2). Once the agents were trained, we compute the accuracy at the last generation-step also on an out-of-sample dataset without uncertainty, i.e. with fixed probabilities, in order to compare the agents on a same out-of-sample.
We present results regrouped in three different situations: training in different world scenarios, training with different numbers of choices and training with different selection rules.

6.1.1 Training in different world scenarios: Accuracy on training and out-of-sample datasets

In this subsection we want to understand how our agents learn according to the worlds in which they evolved. We analyse the following situations:

- Fixing:
  - Number of generations (Ngen=400)
  - Number of choices per lifetime (L=150)
  - Type of selection (Sigmoid-weighted, inflection point=10%, steepness=20)
- Varying
  - The world scenario: SU, U2, U1, WU, FL and CE

The accuracy on the training set and on a test set (out-of-sample built in the same conditions) in function on generations are shown in Figures 6.1 for SU, 6.2 for U2, 6.3 for U1, 6.4 for WU, 6.5 for FL, 6.6 for CE.
In the following Table 6.1, we report the averaged results (for 10 different simulations) of the accuracy on the training set, on an out-of-sample same-scenario of the training set and on an out-of-sample different-scenario (CE world). For example, the accuracy of the NN trained in the world SU and tested with CE is 0.86, while the accuracy of the NN trained in the world FL and tested with an out-of-sample FL different from the training one is 0.92. The accuracy is here computed after 500 generations, with the same parameters listed before.

	SU	U2	U1	WU	FL	CE
Training	0.98	0.89	0.79	0.74	0.94	0.92
Same-world out-of-sample	0.87	0.80	0.76	0.82	0.92	0.90
CE out-of-sample	0.86	0.84	0.72	0.61	0.88	0.90

Table 6.1: Accuracy on training set, out-of-sample same scenario and out-of-sample different scenario for the different worlds (Ngen=500, sigmoid-weighted selection infl=10%, L=500)

As expected, the accuracy on the training set is the highest one for each world-scenario, because agents actually have learnt how to make choices on that dataset. On the other hand, the accuracy on the out-of-sample built in a different scenario (CE world) is the lowest one between the three measures of accuracy (on training set, out-of-sample same scenario and out-of-sample different scenario), and this can be explained by the fact that agents trained in different scenarios never faced choices in which the probabilities are fixed.
Despite this general trend, we can still notice differences between the worlds: looking into the table, the world scenarios in which agents outperform (beside CE that obviously is the simplest world and the scenario for which the last out-of-sample analysis is the same of the training), are SU and FL. This can be explained from the fact that in these two worlds the agents are trained in difficult and different situations, so they learnt a versatile strategy. This led to success also in a scenario different from the training one, as the CE world.
On the contrary, the out-of sample CE test is worst when agents are trained in world scenarios WU and U1. This can be due to the fact that in these two cases the agents are trained in worlds in which bad outcomes are not existent, hence they can have big difficulties to understand how to make a reasonable choice as soon as more risk appears. Moreover, when

faced to the CE scenario, where bad outcomes are possible, they are put in a completely new situation and they are not able to react.
Generally speaking, when faced to the same out-of-sample with fixed probabilities, agents trained in simplified scenario (CE) will take more optimal decisions compared to agents trained in complex scenario (SU, FL). This can be a first proof of our initial hypothesis, i.e. the fact that agents trained in complex scenario developed an heuristics which will not work optimally in over-simplified scenarios. However, these results are necessary but not sufficient to assume that our hypothesis is correct: one can conclude that the second type of agents (trained with fixed probabilities) perform better just because they are tested in the same training condition, contrary to respect to the first type of agents (trained with uncertain probabilities).

6.1.2 Training with different numbers of choices: Accuracy on training set

In this subsection we want to understand how our agents learn according to the number of choices per lifetime that they face in the training phase. We analyse the following situations:

- Fixing:
  - Number of generations (Ngen=400)
  - The type of scenario: SU and FL
  - Type of selection (Sigmoid-weighted, inflection point=10%, steepness=20)
- Varying:
  - Number of choices per lifetime: 50 choices, 100 choices, 200 choices, 300 choices and 400 choices.

The accuracy on the training set, is shown in Figure 6.7 for the SU scenario and in Figure 6.8 for the FL scenario.
As we can see from the two graphs, in the early stage of the training (around 100 generations) there is a significant difference between the accuracy curves: the agents that face less choices per lifetime misperform compared to the ones who face more choices. This difference tends to disappear as the number of generations increases: after 400 generations, one can not see any difference between the curves.
For this reason, we decide to keep the number of generations high, in order to be sure that the system had the time to stabilize whatever the training set, but to reduce the number of choices per lifetime, to avoid possible over-fitting of the data. The usual dimension of the training dataset used in general in the simulations was then 150 choices per lifetime.

6.1.3 Training with different selection rules: Accuracy on training set

In this subsection we want to understand how our agents learn according to the selection rule applied in the evolutionary algorithm. We analyse the following situations:

Figure 6.1: Accuracy on training set and on test set, both built according to a SU (Super Uncertain) scenario

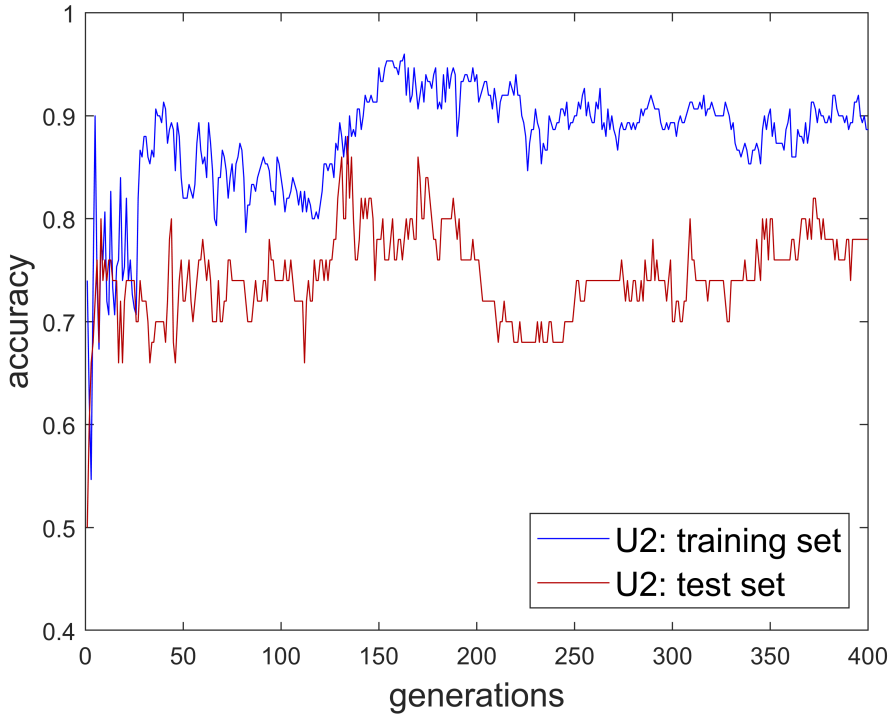


Figure 6.2: Accuracy on training set and on test set, both built according to a U2 (Uncertain-type2) scenario

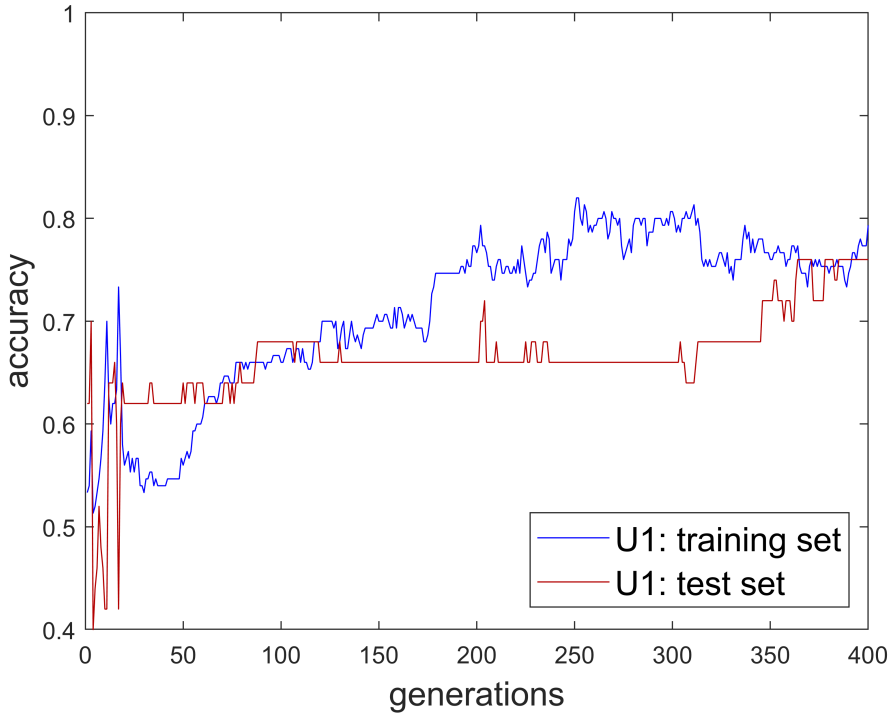


Figure 6.3: Accuracy on training set and on test set, both built according to a U1 (Uncertain-type1) scenario

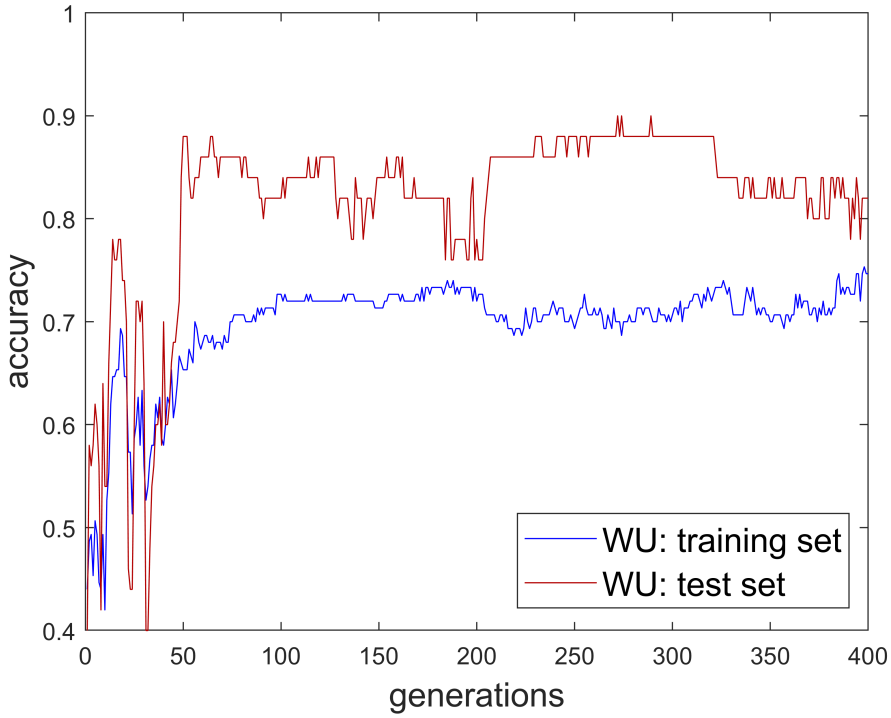


Figure 6.4: Accuracy on training set and on test set, both built according to a WU (Weak Uncertain) scenario

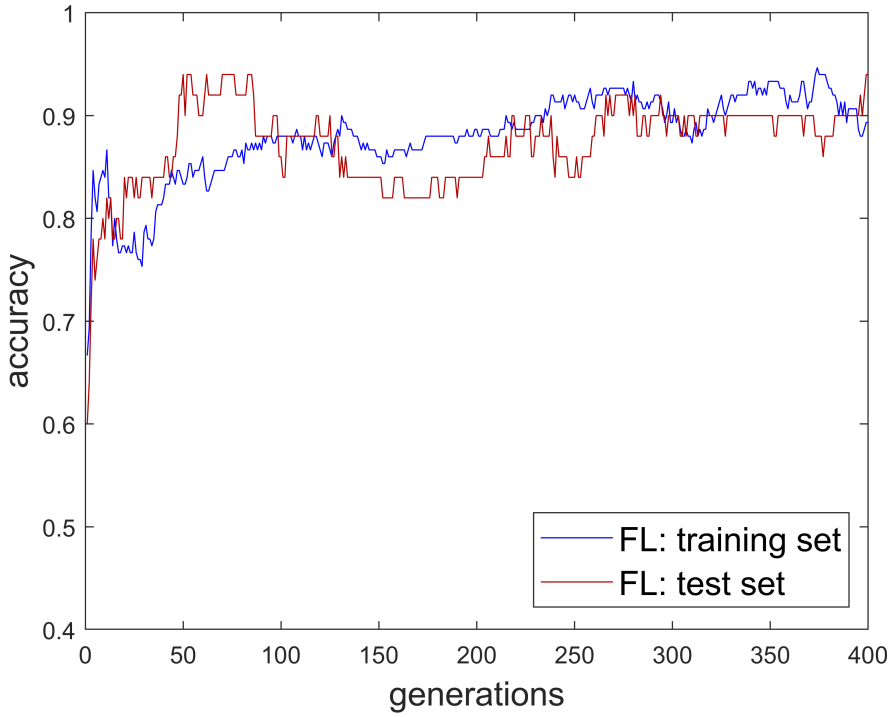


Figure 6.5: Accuracy on training set and on test set, both built according to a FL (Floating) scenario

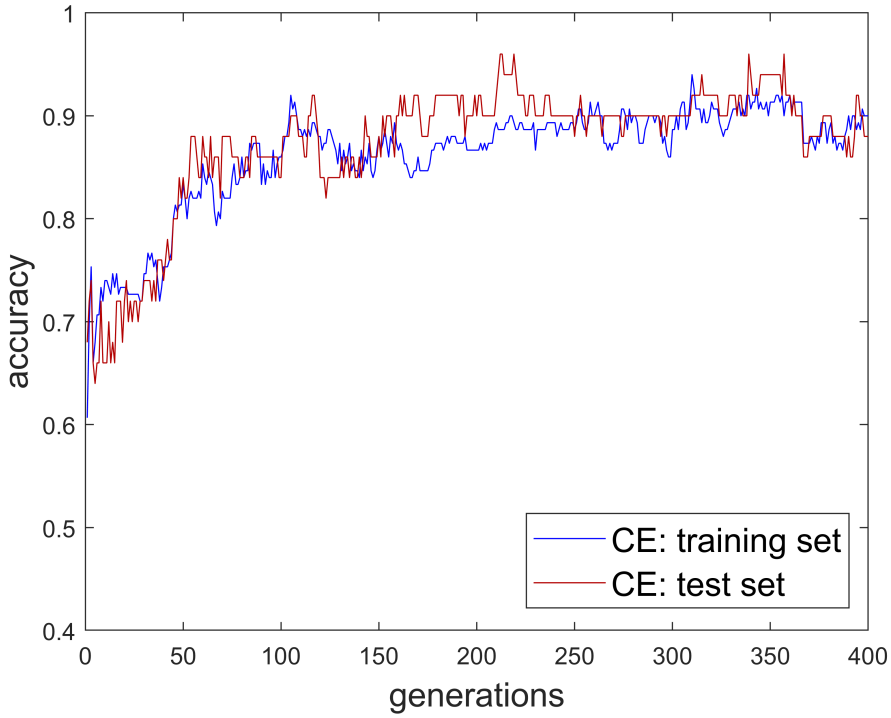


Figure 6.6: Accuracy on training set and on test set, both built according to a CE (Certain) scenario

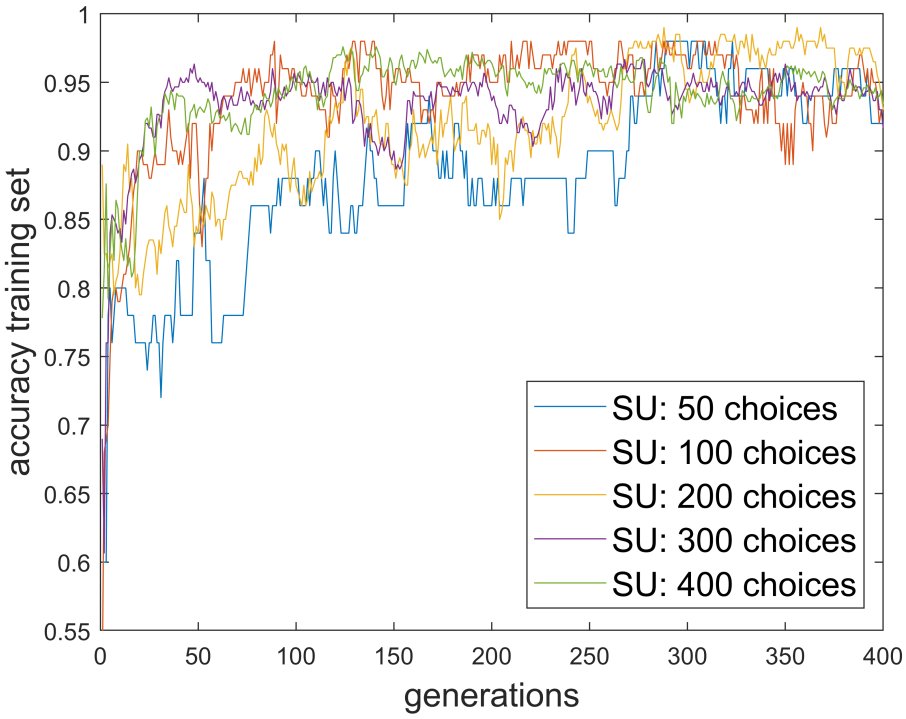


Figure 6.7: Accuracy on training set, built according to a SU (Super Uncertain) scenario for different length of training set, i.e. different number of choices faced to agents in the training phase

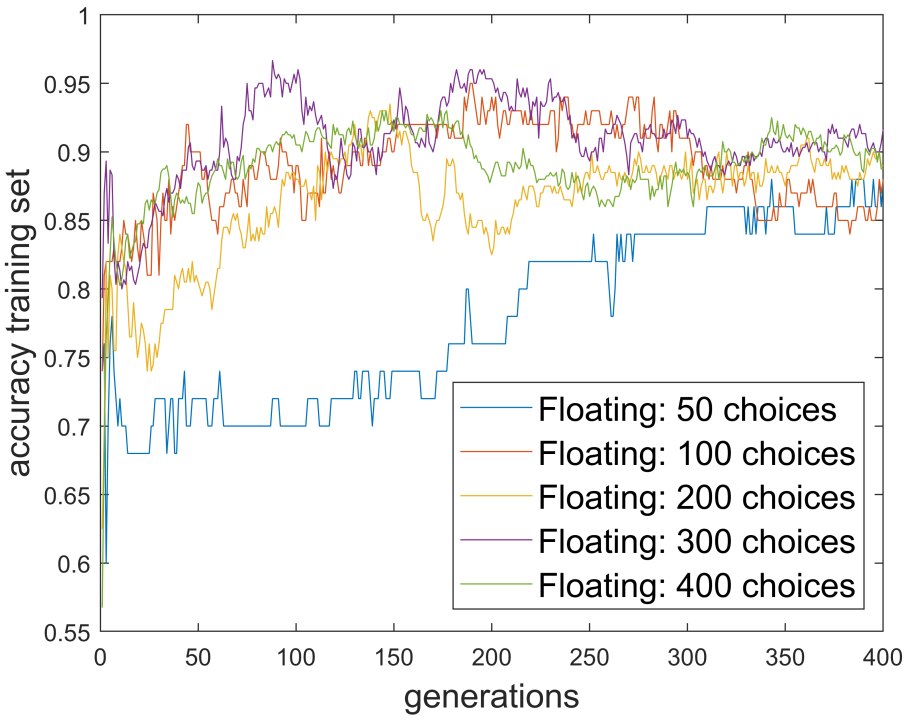


Figure 6.8: Accuracy on training set, built according to a FL (Floating) scenario for different length of training set, i.e. different number of choices faced to agents in the training phase

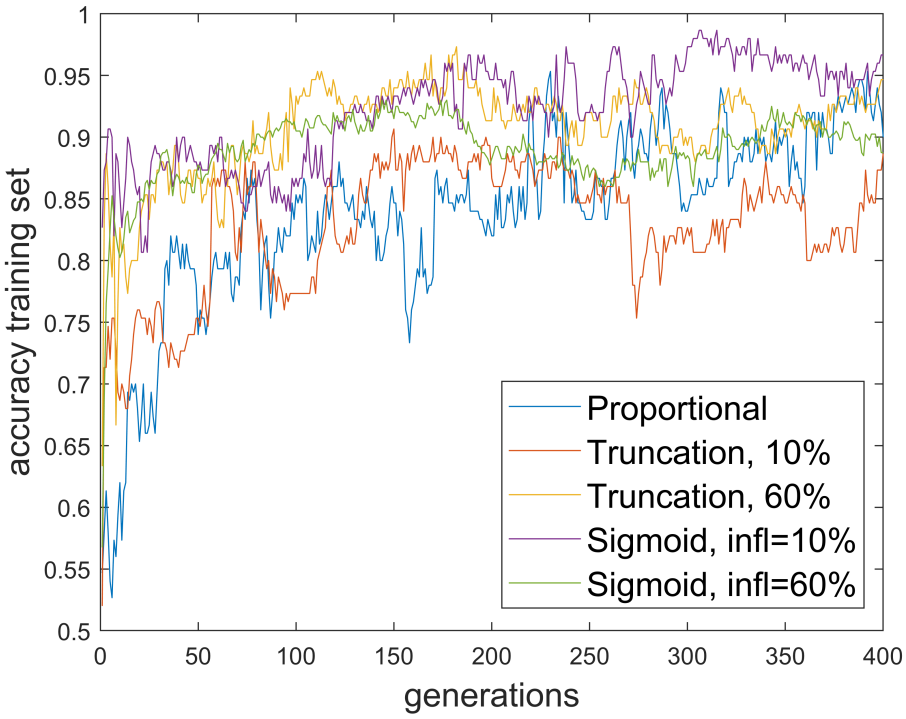


Figure 6.9: Accuracy on the training set for SU world and different selection rules

- Fixing:
  - Number of generations (Ngen=400)
  - The type of scenario: SU
  - Number of choices per lifetime (L=150)
- Varying:
  - Type of selection: proportional, truncation with 10% of survivors, truncation with 60% of survivors, sigmoid-weighted with inflection point at 10%, steepness=10 and sigmoid-weighted with inflection point at 60%, steepness=10

Then we compute the accuracy on the training set as function of generations, as it is shown in Figure 6.9. We can notice that all selection rules brought to a similar accuracy (except truncation 10%). However, the two sigmoid-weighted selection rules give smaller oscillations in accuracy compared to the other selection rules. For this reason, we select this type of rule as default; specifically, we choose as default value of sigmoid inflection point 10%, noticing best results whenever the percentage of surviving agents was small.

6.2 Confront agents with decision theories: Reverse Engineering Analysis

The aim of this section is to find some analogy between our agents behaviour and the one described by some decision theories presented in Chapter 2, specifically by Expected Util-

ity Theory and Rank Dependent Utility Theory.
In order to check for similarities, we first trained the agents with different types of parameters of the ABM (mainly changing the training world-scenario, the number of choices in the training dataset and the type of selection) and then we used the Reverse Engineering procedure described in Section 4.4.
The out-of-sample dataset faced to the agents in this reverse engineering part was always the same: it was built following a CE scenario and it is consists of 50 binary choices. When not specified, we imposed Ngen=500, L=150, sigmoid selection rule with steepness=10 (varying inflection point).

6.2.1 Risk aversion with different selection rules

As explained in Chapter 2, the presence of risk aversion in individuals is characterized by a concave utility function, while a convex utility function is a sign of risk seeking.
Understanding how selection acts on risk preference for our agents is crucial to interpreting and confronting our results with observed individual behaviors. According to different selection rules, individuals can develop different attitudes to risk. In particular, it is reasonable to assume that when an environment is particularly challenging, i.e. only few best individuals will survive, being risk seeker is the best strategy, because only the ones taking risks (and being lucky) will outperform with respect to the ones preferring lower safer outcomes (risk averse). On the other hand, if the selective pressure is less strong, e.g. in a scenario where a large percentage of individuals will survive, the best strategy will be trying to survive with the lowest possible risk, hence it is reasonable to think that the individuals will be risk averse.
In order to asses how attitude towards risk changes according to how challenging the training environment is, we analyse the following situations:

- Fixing:
  - Number of generations (Ngen=500)
  - Number of choices per lifetime (L=150)
  - Type of selection rule: Sigmoid-weighted, steepness=10
- Varying:
  - Type of scenarios: CE, FL, SU
  - Inflection point of the sigmoid: (10%, 20%, 60%, 80%)

We then fit the utility function of our agents applying the Reverse Engineering algorithm, but focusing only on Expected Utility, hence the estimated utility function is found only assuming the expected utility model (objective probabilities).
The results are shown in Figure 6.10 for agents trained in CE scenario, 6.11 for the FL scenario, 6.11 for the SU scenario. As expected, there is a shift from risk seeking towards risk aversion each time the value of sigmoid inflection point is increased, i.e. each time the percentage of surviving agents is increased. From the plots we can indeed observe that, whatever the level of uncertainty in the training phase, if the agents evolved in a challenging environment (only a small percentage of the population survived), when faced

to out-of-sample choices, they tend to be risk seekers, while, if trained in a non-challenging environment (a large percentage of the population survived), they usually behave as risk averse.
From these results, we conclude that risk preferences strictly depend on the particular evolutionary process. Our results are in line with what found by Kolodny and Stern in [Kolodny and Stern, 2017]. This paper study the effects of reproduction dynamics on the extent to which different strategies of risk preference are favored by selection, conducting an agent-based simulation in constant-sized populations. They explore the effects of inter-generational different selection rules, population size, the number of decisions throughout an individual's life, concluding that simple rules regarding predicted risk preference do not hold across the complete range of factors characterizing evolution.

6.2.2 Probability distortion with different uncertainty training scenarios

As we saw in Chapter 2, individuals are usually bad in understanding and interpreting probabilities. Studying how humans tend to distort probabilities, Tversky and Kahneman [Tversky and Kahneman, 1992] first proposed a probability weighting function to convert objective probabilities into subjective ones. The usual form adopted for the probability weighting function is shown in Figure 2.2, and it accounts for over-weighting of small probabilities and under-weighting of large ones.
When trained in different scenarios (with different level of uncertainty), agents could perceive probabilities differently, leading to distortion of probabilities. In order to asses how individuals may weight probabilities according to the uncertainty level in the training phase, we analyse the following situations:

- Fixing:
  - Number of generations (Ngen=500)
  - Number of choices per lifetime (L=150)
  - Type of selection rule: Sigmoid-weighted, steepness=10
- Varying:
  - Training world-scenario
  - Sigmoid inflection point

Here we search for probability weighting, so we apply the Reverse Engineering algorithm assuming only Rank Dependent Utility theory (which includes also Expected Utility considering a $w(p) = p$). As explained in Section4.4, there are two possible probability weighting functions that can be used to fit the agents behaviors: equation $w_1(p)$ 4.5, with just one parameter δ and equation $w_2(p)$ 4.6, with two parameters γ and κ. We performed a reverse engineering analysis for both weighting functions, however we noticed that the fitted curves were similar in pattern, as can we observe from Figure 6.13. For this reason, we present the results just for the probability weighting function $w_1(p)$, being more parsimonious.

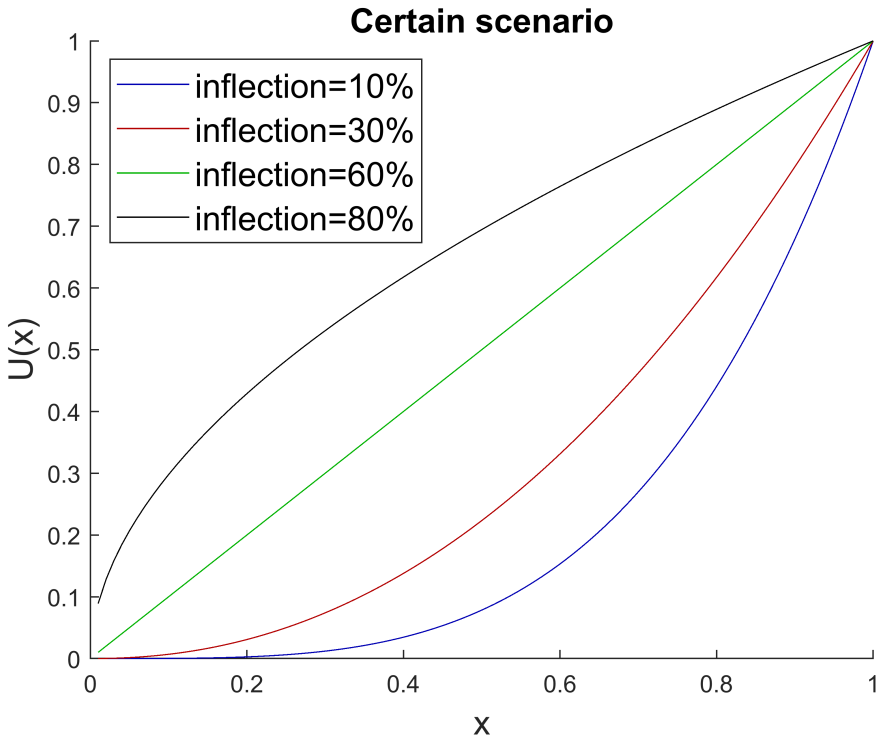


Figure 6.10: Fitted curves of utility function for CE scenario. Fitted values of parameter of α are: $\alpha = 3.67$ for sigmoid with inflection 10%, $\alpha = 2.16$ for sigmoid with inflection 30%, $\alpha = 1.00$ for sigmoid with inflection 60%, $\alpha = 0.53$ for sigmoid with inflection 80%

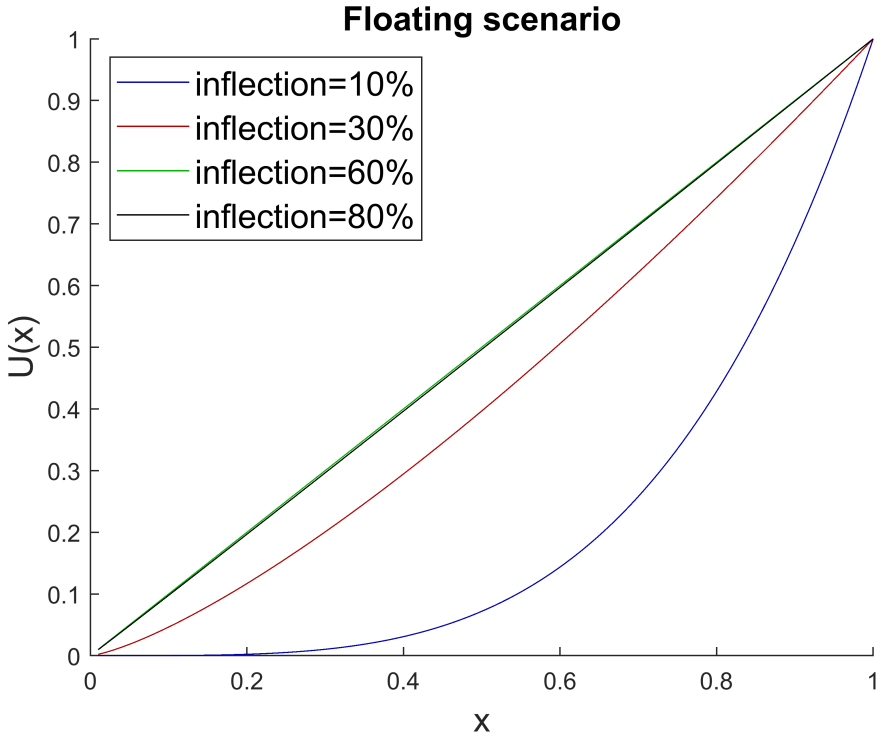


Figure 6.11: Fitted curves of utility function for FL scenario. Fitted values of parameter of α are: $\alpha = 3.79$ for sigmoid with inflection 10%, $\alpha = 1.33$ for sigmoid with inflection 30%, $\alpha = 1.00$ for sigmoid with inflection 60%, $\alpha = 1.01$ for sigmoid with inflection 80%

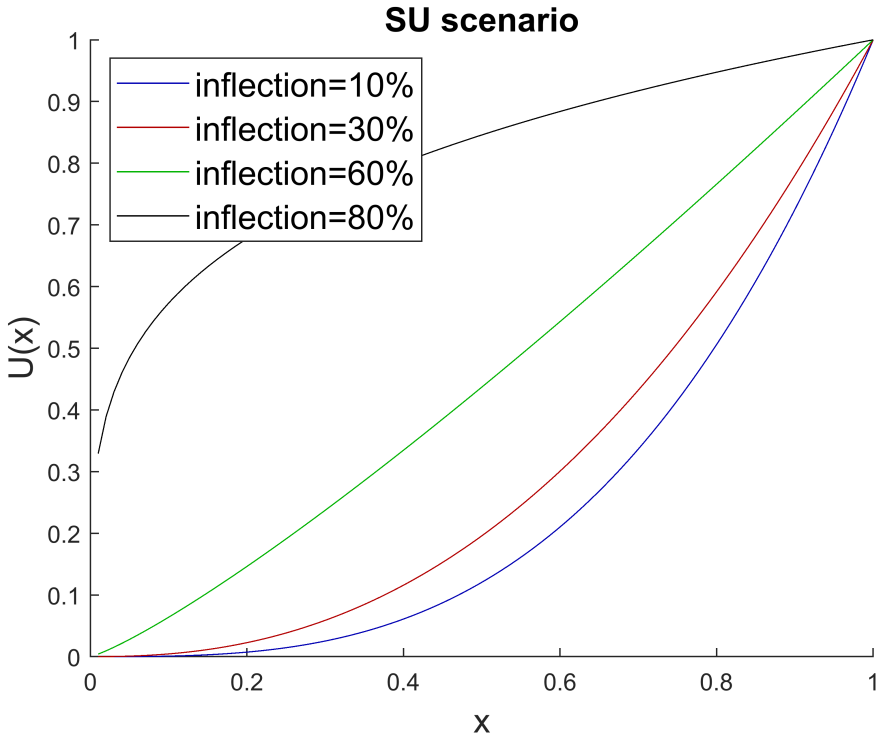


Figure 6.12: Fitted curves of utility function for SU scenario. Fitted values of parameter α are: $\alpha = 3.05$ for sigmoid with inflection 10%, $\alpha = 2.35$ for sigmoid with inflection 30%, $\alpha = 1.19$ for sigmoid with inflection 60%, $\alpha = 0.25$ for sigmoid with inflection 80%

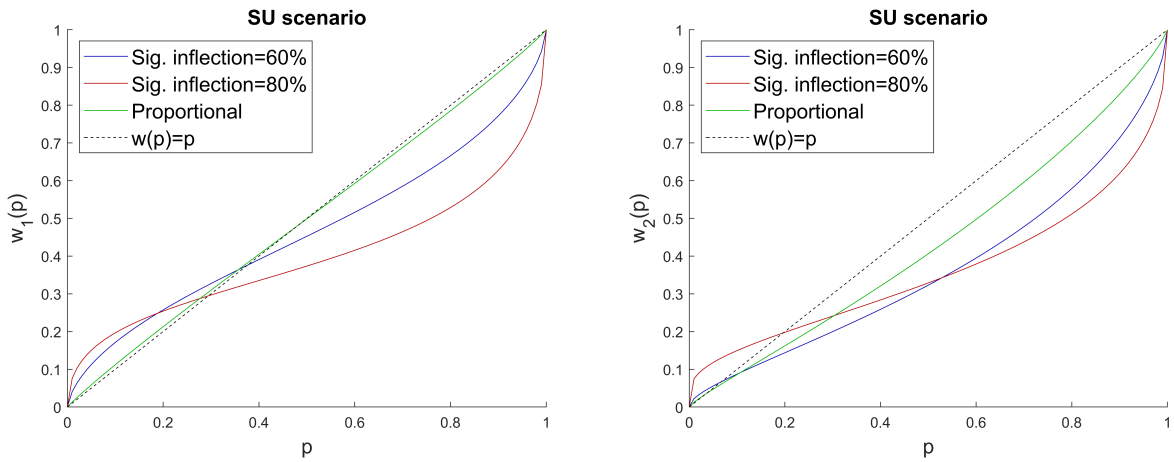


Figure 6.13: Fit of same agents outcomes to two different probability functions $w_1(p)$ and $w_2(p)$, for SU scenario and different inflection points

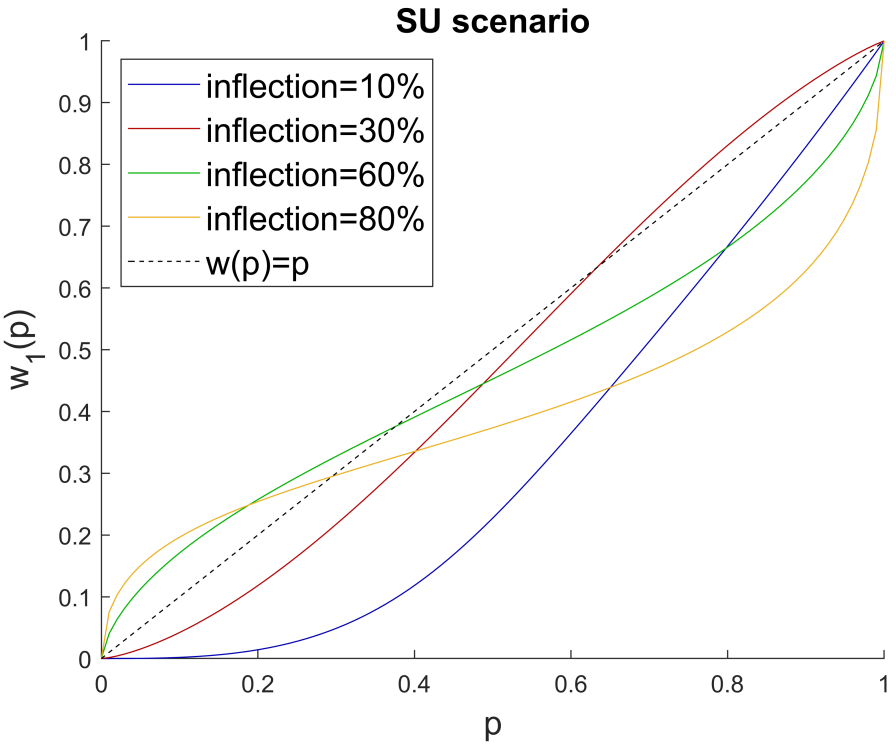


Figure 6.14: Probability weighting SU scenario for different sigmoid inflection points. Fitted values of parameter δ are: $\delta = 2.77$ for sigmoid with inflection 10%, $\delta = 1.40$ for sigmoid with inflection 30%, $\delta = 0.68$ for sigmoid with inflection 60%, $\delta = 0.52$ for sigmoid with inflection 80%

Before looking into the different world scenarios and how they affect the probability weighting, let us focus on the effect of the sigmoid inflection point, as shown in Figure 6.14. Interestingly, it seems that the less the environment is challenging (more individuals are able to reproduce) the more the probability weighting is accentuated towards an over-weighting of low probabilities and an under-weighted of high probabilities, which is the pattern proposed by [Tversky and Kahneman, 1992]. The curve fitted with a sigmoid-weighted selection with inflection point 60% has a parameter $\delta = 0.68$, extremely similar to the one fitted in real experiments [Barberis, 2013].
On the other hand, where the environment is very challenging (only 10% or 30% of the best individuals will reproduce) we can observe an under-weighting of low probabilities and, just for an inflection point 30%, an over-weighting of high probabilities, in contrast with Prospect Theory.
In order to compare the different world scenarios used in the training phase, we report the different fitted curves for a sigmoid-weighted selection rules with inflection point at 80%, Figure 6.15.
As one can see, the two words with a stronger probability weighting are SU and FL, and this is reasonable thinking that these two worlds are characterized by more complex and uncertain choices. On the other hand, for the worlds WU and U1 we can see that there is no probability weighting, i.e. the fitted probability functions are coincident (or almost coincident for U2) with the diagonal dotted line, corresponding to linear probability weighting.

An extreme probability weighting is sign of a strong distortion of probability perception, and this can lead to irrational decisions. The above described findings are thus in line with our first hypothesis that heuristics developed in complex environment can lead to paradoxical results when applied to over-simplified scenarios.
In the previous analysis on accuracy, we found that the selection rule that best mimics (i.e. that gives higher accuracy and less oscillations) the actual evolution process for our agents is the sigmoid-weighted selection with a low percentage of survivors (e.g. inflection point 10%).
The comparison in probability weighting between the different scenarios with such selection rule is shown in Figure 6.16. As we can see, the corresponding fitted probability weighting function is different from the one inferred by Prospect Theory (Section 2.3), involving an under-weighting of small probabilities and sometimes over-weighting of high probabilities, as already pointed out in Figure 6.14.
One argument in favour of such result is the so-called "description-experience gap" [Barron and Ursino, 2013]: while rare events are over-weighted in description based decisions (as described in Prospect Theory), people seem to underweight rare events when they make choices in decisions based on experience under uncertain conditions. Evidence on experience-based decision-making, where people do not know the outcome probabilities a priori but can learn them through repeated experience, shows that people consistently choose as if they underweight rare events ([Camilleri and Newell, 2011]; [Lejarraga and Gonzalez, 2011]; [Newell and Rakow, 2007]; [Yechiam and Busemeyer, 2006]). This can lead to paradoxical inconsistency of people's beliefs and choices [Szollosi et al., 2019]. For this reason many scientists, among who Barron and Erev [Barron and Erev, 2003] and Hertwig, Barron, Weber and Elke [Hertwig et al., 2004], argue that such findings are substantive and call for a theory of decision making under risk, other than Prospect Theory, for explaining decisions from experience.
In view of the above described scientific researches, the probability weighting shown in Figure 6.16 could be meaningful, and it should be deeper studied and analyzed.

6.2.3 Method comparison with different scenarios

In order to compare Expected Utility Theory and Rank Dependent Utility Theory, we used the Nested Hypothesis Testing as explained in Section 4.4. We choose as significant level $\epsilon = 0.01$ and as probability weighting function $w_1(p)$ 4.5, meaning that the degree of freedom of the chi-squared distribution used by the Nested Hypothesis Testing is $k = 1$ (i.e. Rank-Dependent model has only one parameter more than Expected Utility).
We remember that we reject the null hypothesis, i.e. EU model is the true one, if

$$p = P(\chi^2(k=1) \geq 2log\frac{\mathcal{L}_{RDU}}{\mathcal{L}_{EU}}) < \epsilon$$

We perform this analysis for a population of agents trained in different worlds and following a sigmoid-weighted selection rule with inflection point 10% and 60%. The nested hypothesis testing is done on a same out-of-sample set, i.e. we are trying to understand which theory best describes our agents when faced with overly-simplified choices, depending on the way they were trained. We collect the following results, represented in Table 6.2.

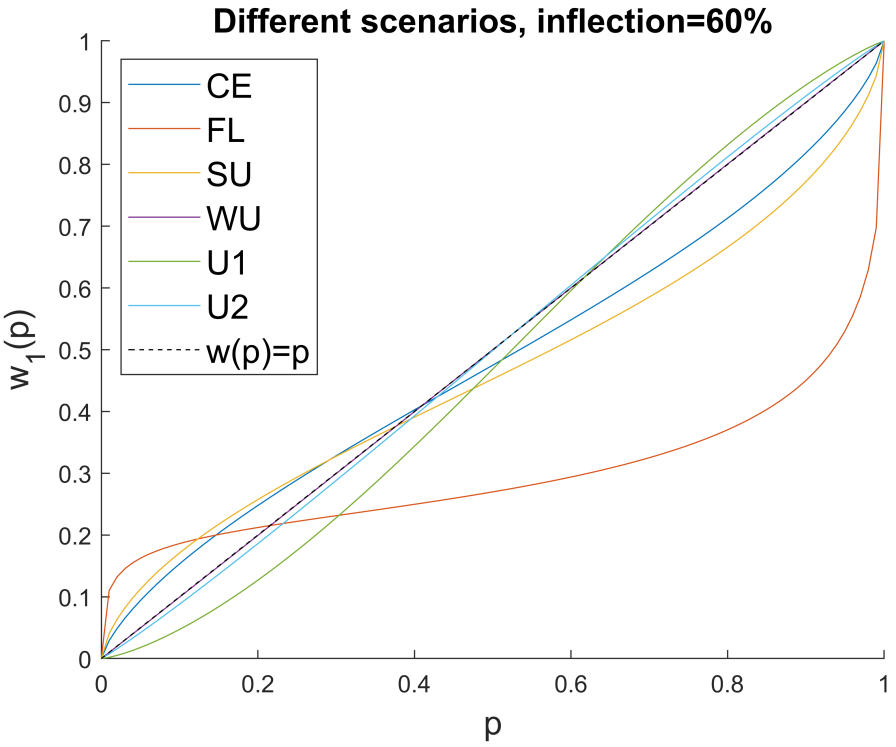


Figure 6.15: Probability weighting different scenarios, sigmoid inflection point=60%. Fitted values of parameter δ are: $\delta = 0.76$ for CE, $\delta = 0.40$ for FL, $\delta = 0.68$ for SU, $\delta = 1.00$ for WU, $\delta = 1.35$ for U1, $\delta = 1.06$ for U2

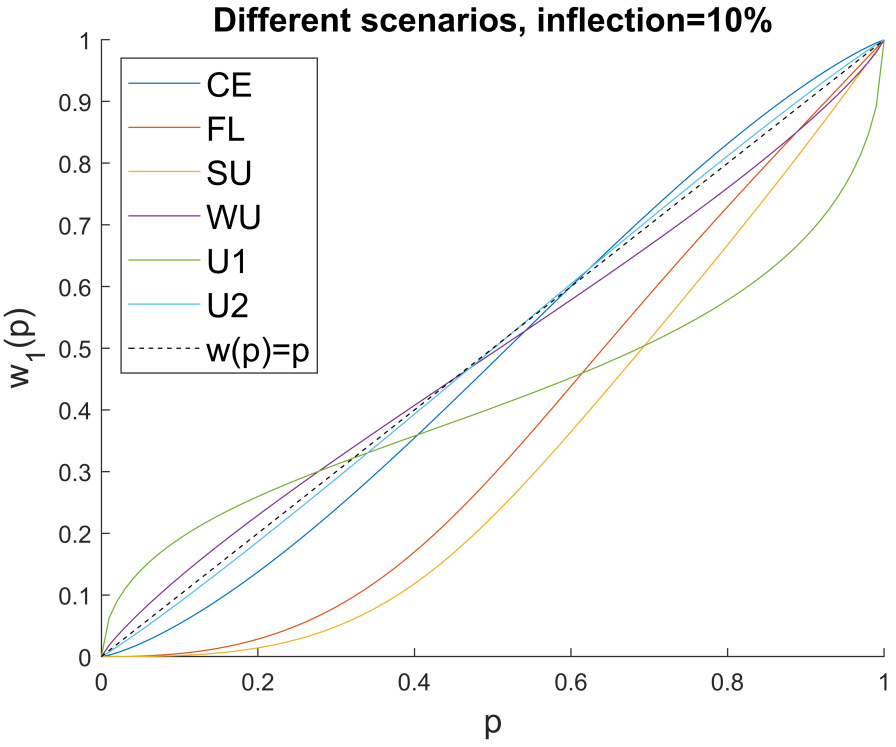


Figure 6.16: Probability weighting different scenarios, sigmoid inflection point=10%. Fitted values of parameter δ are: $\delta = 1.29$ for CE, $\delta = 2.34$ for FL, $\delta = 2.77$ for SU, $\delta = 0.86$ for WU, $\delta = 0.57$ for U1, $\delta = 1.06$ for U2

Sigmoid inflection point 10%	SU	U2	U1	WU	FL	CE
Winning model	RDU	EU	RDU	RDU	RDU	RDU
Corresponding p	0	0.1177	0	0.0010	0	5.7418e-11
Sigmoid inflection point 60%						
Winning model	RDU	EU	RDU	RDU	RDU	EU
Corresponding p	0.0012	0.2530	3.5220e-05	0.0040	0	0.0953

Table 6.2: Nested hypothesis testing: p-value and best models

The right model is almost always Rank Dependent Utility, which is reasonable since RDU is a more complete theory, described by one parameter more that EU. Despite that, one can notice that in general the p-values for the inflection point of 60% are higher with respect to the ones related to the inflection point of 30%.
Moreover, we observe differences among the agents trained in different uncertainty settings: the p-values related to the CE scenario are higher with respect to the ones related to SU and/or FL scenario, implying that the EU model works better for CE then for SU and/or FL. In simpler words, when agents were evolved in an extremely uncertain environment, RDU outperform EU more than for agents trained in simpler environment. This is reasonable since the CE world is a simple scenario, characterized by fixed probabilities, while SU and FL are complex and uncertain scenario, where agents will tend to distort more probabilities, so a theory including a probability weighting is necessary.

6.2.4 Limits of the analysis and possible improvements

Despite these first results are promising, they have just to be intended as a starting point for a more sophisticated analysis.
In view of a possible continuation of this work, we suggest the following amendments and/or improvements:

- switch to a quicker programming language, as Python or Julia, instead of MATLAB
- optimize the code, for example adding some automated strategy to cross-combine and select the best parameters
- explore the possibility of evolving the neural network not just in weights and biases, but in their entire structure, as explained in Section 4.3
- explore different uncertainty set-ups, for example including the uncertainty also in payoffs other than probabilities
- explore other evolutionary algorithm set-ups. In this work we used non-overlapping generations and just started to explore a more realistic overlapping generations scenario; an extension on this sense can be further explored

7. Conclusions and possible extensions

In this thesis, we have presented a simple evolutionary model to study and analyse our decision making processes. Within the field of choice under uncertainty, it is experimentally observed that people tend to make irrational and/or controversial choices (e.g. Allais Paradox [Allais, 1953], Ellesberg Paradox [Ellsberg, 1961]), adopting simple heuristics rather than following rational principles established by expected utility frameworks. Heuristics, which we use everyday to take decisions, work optimally in a real-world-complexity scenario, but only sub-optimally in abstract and oversimplified laboratory setups, leading to the emergence of irrational choice patterns and paradoxes.
This project started from the simple intuition that, since evolution has shaped our decision making processes and heuristics, such observed irrational choice preferences can be retrieved as an evolutionary emergent phenomenon.
To assess our hypothesis, the first phase of the project was dedicated to review the main research behind decision making, studying the most famous theories, as Expected Utility Theory [Morgenstern and Von Neumann, 1953] and Prospect Theory [Kahneman and Tversky, 1979], some fundamentals concepts, as risk aversion and probabilities weighting, and analysing a set of paradoxes and fallacies. Right after, we deepened the concept of heuristics, understanding how heuristics can interfere, and sometimes help, in the process of decision making, mainly analysing the research of Simon [Simon, 1947] and of Gigerenzer [Gigerenzer and Todd, 1999].
After this preliminary research phase, we developed an Agent Based Model (ABM) describing the evolution of the decision making processes. In the model, agents are first trained by facing them with different degrees of uncertainty, with the aim of naturally facilitating the development of heuristics and strategies. Once the training phase has ended, we confronted them with different kind of tasks, to observe if some empirically reported patterns would arise.
In the development of the ABM we brought the most innovative contribution to the project; first of all, to include intelligence in our model, we chose as agents Artificial Neural Networks, exploiting the big potential of this machine learning tool and making our ABM truly innovative. Secondly, we trained the agents by means of an evolutionary algorithm, to mimic the actual Darwinian selection. Choosing such algorithm gives us the double advantage of using a more realistic algorithm, naturally emulating what happens in real-world, and of obtaining quick result, finding in a relatively short time solution-patterns to a problem otherwise hardily solvable with other standard algorithms [Soni, 2018]. Our developed evolutionary algorithm is composed by different steps and characterized by a huge number of parameters (14). Managing and combining all these parameters was one of the biggest challenge of this master thesis.
Finally, in the Reverse Engineering part, we use our statistical knowledge, implementing a

maximum likelihood estimation, to find which decision theory, among the ones analysed during the research phase, best fit the behaviour of the agents, depending on the different training scenarios.
As results, after having verified the functioning of our ABM and carefully selected the best model-parameters to mimic the actual evolution of the agents (thanks to an analysis based on the accuracy with which agents took decisions), we observed different decision-making attitudes, depending on the degree of risk and uncertainty of the choices faced by the agents in the training evolutionary phase. Specifically, to evaluate our agents' performances, we face them with a same out-of-sample of choices characterized by no-uncertainty (known probabilities), and we compare the behaviour of the evolutionary selected agents to the decision making theories of Expected Utility and Rank Dependent Utility.
We first try to understand how selection acts on risk preferences. With this aim, we fit the utility function of our agents assuming Expected Utility Theory, finding that, whatever the uncertainty in the evolutionary process, when the agents evolved in a challenging environment (only a small percentage of the population survive), if faced to out-of-sample choices, they tend to be risk seekers, while, when trained in a non-challenging environment (a large percentage of the population survive), they usually behave as risk averse. From these results, we conclude that it does not exist a straightforward rule to determine an individual risk attitude, but risk preferences strictly depend on the process of evolution. Our results are in line with what found by [Kolodny and Stern, 2017].
Secondly, we analyse how agents perceive and understand probabilities, by fitting the probability weighting assuming Rank Dependent Utility Theory. We found that agents evolved in complex and extremely uncertain environments are subjected to a stronger probability weighting with respect to agents evolved in less or even non uncertain environments. A strong probability weighting is sign of a strong distortion of probabilities perception, leading to irrational choice-patterns; these findings are thus in line with our first hypothesis that heuristics developed in complex environment may lead to paradoxical results when applied in over-simplified scenarios.
In the probability weighting analysis, for non-challenging environments we recover the probability weighting function proposed by Prospect Theory [Tversky and Kahneman, 1992]. On the other hand, for challenging environments, we recover agents under-weighting rare events and sometimes over-weighting probable events. This results, despite being in contrast with Prospect Theory, is supported by other research and experimental results (as [Barron and Erev, 2003], [Camilleri and Newell, 2011], [Newell and Rakow, 2007]), and it should be further explored.
Finally, we found that the best theory between EU and RDU to model our agents behaviour was almost always RDU, due to the fact that it is a more complete theory, described by one parameter more than EU. Despite that, we observe differences among the agents trained in scenarios different in uncertainty, indeed, when agents evolved in extremely uncertain environments, RDU outperform EU more than for agents trained in simpler environments.
In light of such results, possible extensions of the present work can be outlined, in order to gain a deeper understanding of how decision making works. Some examples of further development include exploring the possibility of evolving the neural network in their entire structure, as explained in [Frolov, 2018], [Harvey, 2017]; exploring different uncertainty set-ups, for example including the uncertainty also in payoffs other than probabilities; exploring other evolutionary algorithm set-ups, as overlapping generations scenarios.

Bibliography

[Allais, 1953] Allais, M. (1953). Le comportement de l'homme rationnel devant le risque: critique des postulats et axiomes de l'école américaine. *Econometrica: Journal of the Econometric Society*, pages 503–546.

[Barberis, 2013] Barberis, N. C. (2013). Thirty years of prospect theory in economics: A review and assessment. *Journal of Economic Perspectives*, 27(1):173–96.

[Barron and Erev, 2003] Barron, G. and Erev, I. (2003). Small feedback-based decisions and their limited correspondence to description-based decisions. *Journal of Behavioral Decision Making*, 16(3):215–233.

[Barron and Ursino, 2013] Barron, G. and Ursino, G. (2013). Underweighting rare events in experience based decisions: Beyond sample error. *Journal of Economic Psychology*, 39:278–286.

[Bernoulli, 1738] Bernoulli, D. (1738). Originally published in 1738; translated by dr. lousie sommer.(january 1954)." exposition of a new theory on the measurement of risk. *Econometrica*, 22(1):22–36.

[Bernoulli, 1954] Bernoulli, D. (1954). Exposition of a new theory on the measurement of risk. *Econometrica*, 22(1):23–36.

[Bonabeau, 2002] Bonabeau, E. (2002). Agent-based modeling: Methods and techniques for simulating human systems. *Proceedings of the national academy of sciences*, 99(suppl 3):7280–7287.

[Câmara, 2015] Câmara, D. (2015). Evolution and evolutionary algorithms.

[Camilleri and Newell, 2011] Camilleri, A. R. and Newell, B. R. (2011). When and why rare events are underweighted: A direct comparison of the sampling, partial feedback, full feedback and description choice paradigms. *Psychonomic bulletin & review*, 18(2):377–384.

[Cherry, 2020] Cherry, K. (2020). Heuristics and cognitive biases. *Verywellmind*.

[de Montmort, 1713] de Montmort, P. R. (1713). *Essay d'analyse sur les jeux de hazard...* J. Quillau.

[Ellsberg, 1961] Ellsberg, D. (1961). Risk, ambiguity, and the savage axioms. *The quarterly journal of economics*, pages 643–669.

[Ferro et al., 2020] Ferro, G., Kovalenko, T., and Sornette, D. (2020). Estimation and comparison between rank-dependent expected utility, cumulative prospect theory and quantum decision theory. *SSRN*.

[Ferro and Sornette, 2020] Ferro, G. and Sornette, D. (2020). Stochastic representation decision theory: How probabilities and values are entangled dual characteristics in cognitive processes. *Swiss Finance Institute Research Paper*, (20-25).

[Fogel et al., 1990] Fogel, D. B., Fogel, L. J., and Porto, V. (1990). Evolving neural networks. *Biological cybernetics*, 63(6):487–493.

[Frolov, 2018] Frolov, S. (2018). Neuroevolution: A primer on evolving artificial neural networks. *Inovex Blog*.

[Gigerenzer, 2018] Gigerenzer, G. (2018). The bias bias in behavioral economics. *Review of Behavioral Economics*, 5(3-4):303–336.

[Gigerenzer and Brockman, 2003] Gigerenzer, G. and Brockman, J. (2003). Conversation: smart heuristics. *Edge*.

[Gigerenzer and Gaissmaier, 2011] Gigerenzer, G. and Gaissmaier, W. (2011). Heuristic decision making. *Annual review of psychology*, 62:451–482.

[Gigerenzer and Todd, 1999] Gigerenzer, G. and Todd, P. M. (1999). *Simple heuristics that make us smart*. Oxford University Press, USA.

[Gonzalez and Wu, 1999] Gonzalez, R. and Wu, G. (1999). On the shape of the probability weighting function. *Cognitive psychology*, 38(1):129–166.

[Hardesty, 2017] Hardesty (2017). Explained: Neural networks. *MIT Computer Science Artificial Intelligence Lab*.

[Harvey, 2017] Harvey, M. (2017). Let's evolve a neural network with a genetic algorithm. *Coastline automation, posted on April*.

[Hertwig et al., 2004] Hertwig, R., Barron, G., Weber, E. U., and Erev, I. (2004). Decisions from experience and the effect of rare events in risky choice. *Psychological science*, 15(8):534–539.

[Johnson and Busemeyer, 2010] Johnson, J. G. and Busemeyer, J. R. (2010). Decision making under risk and uncertainty. *Wiley Interdisciplinary Reviews: Cognitive Science*, 1(5):736–749.

[Kadane and Winkler, 1988] Kadane, J. B. and Winkler, R. L. (1988). Separating probability elicitation from utilities. *Journal of the American Statistical Association*, 83(402):357–363.

[Kahneman et al., 1990] Kahneman, D., Knetsch, J. L., and Thaler, R. H. (1990). Experimental tests of the endowment effect and the coase theorem. *Journal of political Economy*, 98(6):1325–1348.

[Kahneman and Tversky, 1979] Kahneman, D. and Tversky, A. (1979). Prospect theory: An analysis of decision under risk. *Econometrica*, 47(2):263–291.

[Kennedy, 2006] Kennedy, J. (2006). Swarm intelligence. In *Handbook of nature-inspired and innovative computing*, pages 187–219. Springer.

[Kolodny and Stern, 2017] Kolodny, O. and Stern, C. (2017). Evolution of risk preference is determined by reproduction dynamics, life history, and population size. *Scientific reports*, 7(1):1–13.

[Kovalenko and Sornette, 2018] Kovalenko, T. and Sornette, D. (2018). The conjunction fallacy in quantum decision theory. *Swiss Finance Institute Research Paper*, (18-15).

[Lagarias et al., 1998] Lagarias, J. C., Reeds, J. A., Wright, M. H., and Wright, P. E. (1998). Convergence properties of the nelder–mead simplex method in low dimensions. *SIAM Journal on optimization*, 9(1):112–147.

[Lazarou, 2019] Lazarou (2019). Evolving neural networks. *Towards Data Science*.

[Lejarraga and Gonzalez, 2011] Lejarraga, T. and Gonzalez, C. (2011). Effects of feedback and complexity on repeated decisions from description. *Organizational Behavior and Human Decision Processes*, 116(2):286–295.

[Lichtenstein and Slovic, 1971] Lichtenstein, S. and Slovic, P. (1971). Reversals of preference between bids and choices in gambling decisions. *Journal of experimental psychology*, 89(1):46.

[Mallipeddi et al., 2011] Mallipeddi, R., Suganthan, P. N., Pan, Q.-K., and Tasgetiren, M. F. (2011). Differential evolution algorithm with ensemble of parameters and mutation strategies. *Applied soft computing*, 11(2):1679–1696.

[Morgenstern and Von Neumann, 1953] Morgenstern, O. and Von Neumann, J. (1953). *Theory of games and economic behavior*. Princeton university press.

[Nelson, 2013] Nelson, R. (2013). *Probability, stochastic processes, and queueing theory: the mathematics of computer performance modeling*. Springer Science & Business Media.

[Newell and Rakow, 2007] Newell, B. R. and Rakow, T. (2007). The role of experience in decisions from description. *Psychonomic Bulletin & Review*, 14(6):1133–1139.

[Pruitt and Hoge, 1965] Pruitt, D. G. and Hoge, R. D. (1965). Strength of the relationship between the value of an event and its subjective probability as a function of method of measurement. *Journal of Experimental Psychology*, 69(5):483.

[Quiggin, 1982] Quiggin, J. (1982). A theory of anticipated utility. *Journal of economic behavior & organization*, 3(4):323–343.

[Real, 2019] Real, E. (2019). Using evolutionary automl to discover neural network architectures. *URL https://ai. googleblog. com/2018/03/using-evolutionary-automl-to-discover. html*.

[Samuelson and Zeckhauser, 1988] Samuelson, W. and Zeckhauser, R. (1988). Status quo bias in decision making. *Journal of risk and uncertainty*, 1(1):7–59.

[Schmidt and Zank, 2008] Schmidt, U. and Zank, H. (2008). Risk aversion in cumulative prospect theory. *Management Science*, 54(1):208–216.

[Simon, 1947] Simon, H. A. (1947). *Administrative behavior*. Simon and Schuster.

[Simon, 1955] Simon, H. A. (1955). A behavioral model of rational choice. *The quarterly journal of economics*, 69(1):99–118.

[Simon, 1990] Simon, H. A. (1990). Invariants of human behavior. *Annual review of psychology*, 41(1):7.

[Soni, 2018] Soni, D. (2018). Introduction to evolutionary algorithms. *Towards Data Science*.

[Stanley and Clune, 2017] Stanley, K. and Clune, J. (2017). Welcoming the era of deep neuroevolution. *Uber Engineering Blog Research*.

[Szollosi et al., 2019] Szollosi, A., Liang, G., Konstantinidis, E., Donkin, C., and Newell, B. R. (2019). Simultaneous underweighting and overestimation of rare events: Unpacking a paradox. *Journal of Experimental Psychology: General*.

[Todd and Gigerenzer, 2000] Todd, P. M. and Gigerenzer, G. (2000). Précis of" simple heuristics that make us smart". *Behavioral and brain sciences*, 23(5):727–741.

[Tversky and Kahneman, 1974] Tversky, A. and Kahneman, D. (1974). Judgment under uncertainty: Heuristics and biases. *science*, 185(4157):1124–1131.

[Tversky and Kahneman, 1981] Tversky, A. and Kahneman, D. (1981). The framing of decisions and the psychology of choice. *science*, 211(4481):453–458.

[Tversky and Kahneman, 1992] Tversky, A. and Kahneman, D. (1992). Advances in prospect theory: Cumulative representation of uncertainty. *Journal of Risk and uncertainty*, 5(4):297–323.

[Wilks, 1938] Wilks, S. S. (1938). The large-sample distribution of the likelihood ratio for testing composite hypotheses. *The annals of mathematical statistics*, 9(1):60–62.

[Yechiam and Busemeyer, 2006] Yechiam, E. and Busemeyer, J. R. (2006). The effect of foregone payoffs on underweighting small probability events. *Journal of Behavioral Decision Making*, 19(1):1–16.

[Yukalov and Sornette, 2008] Yukalov, V. I. and Sornette, D. (2008). Quantum decision theory as quantum theory of measurement. *Physics Letters A*, 372(46):6867–6871.

[Yukalov and Sornette, 2011] Yukalov, V. I. and Sornette, D. (2011). Decision theory with prospect interference and entanglement. *Theory and Decision*, 70(3):283–328.