AI Alignment with Artificial Life

A Concrete Approach to the Abstract Problem of Controlling Advanced AI

Master's thesis

Tomas Lundberg
Richard Martin

Department of Mathematics
Chalmers University of Technology
Gothenburg, Sweden 2021

Master's thesis 2021

AI Alignment with Artificial Life

A Concrete Approach to the Abstract Problem of Controlling Advanced AI

Tomas Lundberg
Richard Martin

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Mathematics
Chalmers University of Technology
Gothenburg, Sweden 2021

AI Alignment with Artificial Life
A Concrete Approach to the Abstract Problem of Controlling Advanced AI
Tomas Lundberg
Richard Martin

© Tomas Lundberg, Richard Martin 2021.

Supervisor: Torbjörn Lundh, Department of Mathematics
Examiner: Torbjörn Lundh, Department of Mathematics

Department of Mathematics
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: Paperclips. Referes to the Paperclip Maximizer of Boström [1].

Typeset in LaTeX
Printed by Chalmers Reproservice
Gothenburg, Sweden 2021

AI Alignment with Artificial Life
A Concrete Approach to the Abstract Problem of Controlling Advanced AI
Tomas Lundberg
Richard Martin
Department of Mathematics
Chalmers University of Technology

Abstract

The Alignment Problem is the issue of how to make sure that a hypothetical Artificial General Intelligence (AGI) pursues the same goals as us humans and it is by many considered to be one of the most important that humanity faces [2]. Much of the work on it is theoretical. Therefore, the motivation of this thesis was to explore a novel approach to quantitatively studying the Alignment Problem in a low-level analogy using the artificial life platform Avida. We modelled the artificial life forms, the Avidians, as self-improving AI with their own goals while analogous humans existed in the system, which in turn had their goals. The goals had a straightforward connection to the fitness landscape. We studied different strategies of how to align the goals of the Avidians with those of the humans. One of the tools we used to obtain aligned strategies was a genetic algorithm on top of the Avida evolution, creating a meta evolutionary system. Our findings are in line with what is considered to be three key factors in creating aligned AGI; that the AI's should be passive if uncertain about what to do, clear communication should be kept with humans and self-tinkering with rewards should be kept under control. Our findings show that our low-level analogous system can indeed be used to study the Alignment Problem, even though it loses a lot of the relevant complexity of the Alignment Problem. Nevertheless our findings show that some key aspects are captured. We hope that our concrete and quantitative system can inspire future studies and be incorporated into the large toolbox that is needed to solve the Alignment Problem.

Keywords: AI-safety, the Alignment Problem, the Problem of Control, AGI, Avida, Artificial Life, Evolutionary Algorithms

Acknowledgements

We would, first of all, extend our utmost gratitude to our eminent supervisor and examiner professor Torbjörn Lundh. He is the creator of the thesis proposal and without his knowledge and valuable input, this thesis would not have been possible. Torbjörn has been an instrumental source of ideas and has provided impeccable support throughout our thesis.

Furthermore, during the course of the thesis, we have had meetings with professor Olle Häggström and associate professor Philp Gerlee. They are true experts in the field of artificial general intelligence and artificial life, respectively, and their inspiring feedback has been invaluable.

Additionally, we are indebted to professor Charles Ofria and the many other contributors of the impressively versatile artificial life platform Avida. Without it, we would not have been able to go through with this thesis.

Moreover, for the preliminary work that had been done by our fellow master student John Harryson, we are exceptionally grateful. His groundwork acted as an advantageous head start on the thesis.

We also acknowledge the fantastic camaraderie and social stimulation that our fellow Masters student friends in the LB society have held up over the years we spent together. It has been of great support for our motivation throughout this project.

Finally, we want to show our sincere appreciation for Chalmers University of Technology and the department of mathematics in particular for providing the opportunity for this thesis. A special thanks go out to professor Martin Raum and professor Stig Larsson for accommodating and allowing us to use the vital computational resources Gantenbein, Hebbe and Vera. Without the indefatigable labour of this number-crunching triad, our simulations would probably still be running...

Tomas Lundberg & Richard Martin, Gothenburg, June 2021

Contents

1
Introduction

> *The first ultra-intelligent machine is the last invention that man need ever make, provided that the machine is docile enough to tell us how to keep it under control.*
>
> - I.J Good 1965 [3]

Artificial intelligence (AI) is already having a large positive impact on our society. It is making businesses more effective, replacing human labour in the execution of monotonous tasks, helping doctors in diagnosing patients, improving education, optimizing transport, enhancing public safety, reducing environmental impact, making driving safer, enhancing the entertainment industry, assisting scientists in improving our understanding of the world, and much more [4]. The impact of AI on our society will only increase as our hardware become better, our theoretical understanding of the systems improve and more data is collected and incorporated into our AI models.

However, the advent of this AI revolution is coupled with many risks and issues. These problems can be divided into short and long term ones. The short term problems include the accumulation of wealth in a small minority, lethal autonomous weapons, loss of jobs, biased AI models and deep fakes among others [5, 6, 7, 8]. In this thesis, we are nevertheless concerned with the long term problems surrounding AI. In the long term, if AI becomes so capable that it matches and exceeds the intelligence of humans in most relevant domains, it might be, as I.J Good[1] [3] put it in 1965, "[...] the last invention that man need ever make [...]" and it could bring about immense wealth and prosperity and according to Torres [9] result in an outcome "[...] not merely *good* but genuinely *utopian*"[2]. However, this will only be the case if we can guard our human interests (see the full quote of I.J Good above). To have intelligence one has to have goals,[3] and if not constructed carefully, an AI that is more intelligent than humans, an *Artificial General Intelligence* (AGI), might pursue other goals than ours. Being able to handle the possibility that an AGI has different goals than us is called *the Problem of Control*. The importance of solving it and making sure that AGI is beneficial for humans is agreed upon by Stephen Hawking, Nick Bostrom, Elon Musk, Steve Wozniak, Max Tegmark, Stuart Russel, and many, many more [2].

The importance of the problem of control is also matched by its difficulty to solve. Torres [9] considers it to be "[...] one of the most formidable, high-stakes problems that humanity has ever had to confront". By construction, we only have one chance

[1] Cheif statistician in Alan Turing's code-breaking team in World War II.
[2] Italic in original.
[3] See Section 2.1.2.

to solve the Problem of Control and that is *before* an AGI is created. Additionally, many of the issues surrounding the problem of control are difficult to study without an actual AGI and we are hence in a catch-22.

In this thesis, we adapt the artificial life platform Avida to model some aspects of the Problem of Control. We create a low-level model where analogues of humans and AI's exist with independently controlled goals. With this thesis, we hope to inspire quantitative research and simulations in the area of AI safety.

We begin with introducing the reader to the Problem of Control, AGI and alignment in Chapter 2. We also discuss the underlying motivation of using the artificial life platform Avida and provide a short introduction to how it operates. Thereafter, in Chapter 3, we present our core experimental framework where of studying the Problem of Control. Furthermore, the layout of our simulations is also described in this chapter. The results of these simulations are then shown in Chapter 4 and in turn discussed in Chapter 5. Finally, we conclude the thesis in Chapter 6.

2
Background

2.1 Setting the stage

There are many who doubt the possibility of AGI and by that deny the urgency of solving the Problem of Control [4, 10, 11, 12]. In this section, we will as straightforward as possible address why we believe that this doubt is unfounded and that solving the Problem of Control should indeed be considered of utmost priority.

We refer to an artificial general intelligence (AGI) as

> **Artificial General Intelligence (AGI)**: An artificial intelligence that is at least as capable as humans in all relevant domains[1].

To clarify this definition, take chimpanzees as an example. Chimpanzees do, for instance, actually outperform humans in tasks related to working memory [13]. However, we humans are still more capable in all *relevant* domains, which results in us controlling the future of chimpanzees. If we humans for some reasons decided that we wanted to get rid of all chimpanzees tomorrow, we could, and there would be nothing the chimpanzees would be able to do about it.

This naturally leads us to the *second-species argument*[2] [14]. Creating AGI, which by definition is more capable than us humans in exerting and maintaining control, will essentially hand over the control of the future of humanity to a second, more powerful *species*; effectively putting us in the situation of the chimpanzees from the example in the previous paragraph. That is, if an AGI is created, we will have lost our power to determine our future. Being able to handle this situation is the Problem of Control:

> **The Control Problem**: Being able to handle, in terms of safeguarding the future of humanity, the possibility of an AGI having different goals than us.

When realizing that aligning the goals of the AGI with ours solves the Problem of Control, we are reducing it to the Alignment Problem[3]:

> **The Alignment Problem[4]:** Making sure that an AGI pursues the goals we want it to pursue.

[1] This includes and exceeds the intelligence of a *Human-level AI* as defined by Bostrom [1].

[2] Russel [6] refers to this as *the Gorilla Problem*.

[3] See Christiano's [15] discussion on Alignment as the solution of the Problem of Control for clarification.

[4] This is in particular the *Outer* Alignment problem as opposed to the *Inner* Alignment Problem. See Section 2.2.2.1 for the distinction.

Having established the definitions and the core problems we now turn to elaborate on the importance of these issues.

2.1.1 The creation of AGI

If we deny the possibility of the creation of an AGI, we are essentially saying that there is no possible configuration of atoms that produce higher intelligence than a human brain. This is exceedingly unlikely given the vast possible number of configurations and the fact that there are reasons to believe that intelligence has not been heavily selected for by evolution [16].

There is one objection to this argument, namely substrate dependence[5]. One might argue that real general intelligence can only be achieved in biological brains. Although there is much we do not know about the brain, there is nothing in our understanding of the natural world that points to general intelligence being confined to exist in "wetware" as opposed hardware.

If we deny substrate dependence and reject the notion of human intelligence being the most intelligent configuration in the space of intelligent minds we can establish that creating an AGI is at least possible. But it being possible does not mean that we will create one. Bostrom's [18] *Technological completion conjecture* states that

> **Technological Completion Conjecture:** If scientific and technological development efforts do not effectively cease, then all important basic capabilities that could be obtained through some possible technology will be obtained.

Given the vast economic incentives of improving technology, and of AI technology in particular, ceasing its development seems exceptionally unlikely. However, this conjecture says nothing about *when* an AGI would be created. It might be the case that it requires tremendous efforts and thousands of years to do. An indication that this is not the case is the results from a 2012/2013 survey [19] where AI researchers were asked to estimate when AGI will be created. Although there is a large variance, the mean of the survey is that an AGI will be created with 50% probability in 2040-2050 and with 90% probability by 2075.

A lot of discredit can be attributed to a survey like this given the poor track record that even experts have on estimating the forthcoming of new technology. However, given the enormous impact a hypothetical AGI might have (the possible extinction of humanity for instance, see next section), even just a small probability that it will happen soon warrants motivating large research efforts into solving the Alignment Problem now. Moreover, even if the creation of the first AGI is far ahead in the future, the motivation for working on solving the Alignment Problem is also a function of how difficult the problem is to solve, not just when the solution will be required. And the problem does seem difficult to solve indeed [20].

[5] As opposed to Substrate *Independence*. See Bostrom [17] for further discussion on this assumption.

2.1.2 The creation of AGI as an existential risk

There are many definitions of intelligence from many different disciplines [21]. One that however seems to have become accepted, at least in the area of AI-research, is the Legg-Hutter [21] definition:

> **Intelligence:** Intelligence measures an agents ability to achieve goals in a wide range of environments.

This definition is consistent with the notion of an AGI being more intelligent than us as defined previously. Moreover, it does indeed seem to capture what we mean by intelligence. The key concepts here are agent, goals and a wide range of environments. The definition implies that it is not possible to be intelligent without having goals. If we agree on this implication, and it is certainly difficult to imagine it being any other way, we should agree that solving the Alignment Problem ought to be a priority from the perspective of reducing existential risk[6]. Because if an AGI has *different* goals than ours it could lead to situations where humans go extinct. Note that this does not imply that an AGI necessarily has to be *evil* or that its final goal would be to kill humans. Take for instance our pet example; the human situation in relation to the chimpanzees. We have caused them tremendous suffering by destroying their habitats and using them as test subjects. This is not because we have anything against the chimpanzees per se, but simply because it has been a side effect of us achieving our goals, namely, to acquire land, gather resources and gain medical knowledge for the betterment of the human condition. That is, our goals have not been aligned with the chimpanzees. The same could be the case for us when an AGI is trying to achieve its own goals if they are not aligned with ours.

A common objection to this is that surely an intelligent enough agent will realize that causing so much harm is the wrong thing to do, no matter the personal gain. However, Bostrom's Orthogonality thesis [23] states that intelligence and goals are *orthogonal* to each other:

> **Orthogonality Thesis:** Intelligence and final goals are orthogonal axes along which possible agents can freely vary. In other words, more or less any level of intelligence could in principle be combined with more or less any final goal.

A typical example that illustrates many of the surrounding concepts of the orthogonality thesis is *the Paperclip Maximiser* [1]. Imagine an agent programmed with the simple final goal of producing paper clips. It would begin to mine the earth to gain material and build a factory for producing paperclips. It will start to optimize this process in order to produce as many paper clips as possible. It would build more factories and expand the mining capabilities. If the agent is intelligent enough, that is if it is capable enough to pursue its final goal, when the most accessible materials have been exhausted it will begin dismantling the earth and even take humans as a source of material for its ever-expanding paperclip production. We would become extinct not because the agent has it as its final goal but because it is an *instrumental goal* of the agent. For most final goals an agent can have, gaining more intelligence,

[6]Bostrom [22] defines an existential risk as: *One where an adverse outcome would either annihilate Earth-originating intelligent life or permanently and drastically curtail its potential.*

more power and more resources is an instrumental goal [7]. Therefore, an agent does not have to be *evil* to take over the world, it might simply be a necessary step along the way of achieving virtually any final goal, even an as simple one as producing paperclips.

2.1.3 Solving the Alignment Problem

We hope these arguments has convinced even the sceptical reader that the possibility of creating an AGI is real and that if, and when we do, it is of paramount importance that our goals are aligned, given that we value the future of humanity. So how *do* we solve the Alignment Problem? The task is by no means an easy one but there are ways forward.

2.2 AGI and the Alignment Problem

In this section, we begin with exploring the modus operandi for creating AGI and then, in particular, aligned AGI. Thereafter, we comment on some fundamental issues with studying alignment and through this lens finally motivate the setup of our thesis.

2.2.1 Approach to creating AGI

Arguably, the reason why humans are generally intelligent is that we can adapt to a wide range of new environments, that is, that we can learn. Unless we want to do a rule-based implementation, which quickly becomes unfeasible even for the simplest of environments, this has to be true for an AGI as well. In the simplest of views, humans learn by getting rewards, in terms of neurotransmitters, when she has done something that can be classified as good in terms of achieving a goal. By obtaining the reward, the neuron connections that caused that particular behaviour are strengthened, and that behaviour is more likely to be used in a similar situation in the future. Inspired by this notion of human learning, a framework of artificial learning has emerged, called *Reinforcement Learning* (RL). Given an *objective*, say, win a game of chess, an agent tries to achieve the objective by following a *policy*[8]. Rewards are given to the agent in terms of how well it fulfils the objective. In this sense, the agent is trying to maximize its rewards by optimizing the objective function.

This basic strategy of reinforcement learning has been used to achieve many of the impressive results of AI agents in recent decades such as self-driving cars and AlphaGo beating Lee Sedol in the game of Go. AI can be considered superhuman in these domains but they are far from being general intelligent. They are what is called, *Narrow AI*[9]. If we would put AlphaGo in any other environment except from

[7] See Bostrom's [23] *Instrumental Convergence Thesis*, a compliment to his Orthogonality Thesis.

[8] Which much of the time is a function realized by an artificial neural network, also inspired by the human brain.

[9] Also referred to as *weak* AI.

playing Go it would be no more intelligent than a rock. To go from narrow AI to general AI[10] is a monumental endeavour, but the progress is nevertheless real.

Even though we can have an agent that efficiently learns to achieve its goal, i.e. to maximize its objective function, the problem of specifying the correct objective remains. This task is much more difficult than it might seem at first sight. There are countless examples where the maximization of objectives leads to perverse outcomes. A good source of these examples is "The surprising creativity of digital evolution" [24]. One example here is when researchers wanted to make agents learn to walk in a simulated environment with friction and gravity. They specified the objective function as rewarding the agents for achieving high speeds to promote the evolution of walking behaviour. What happened was that the agents just built themselves as large towers and then fell to the ground, thus achieving high speeds. Specifying the correct objective is very important to achieve the behaviours one actually want. A law from economics that captures these problems well is Goodhart's Law [25]:

> **Goodhart's Law**: When a measure becomes a target, it ceases to be a good measure.

And as for the case with an aligned AGI, the objective necessarily needs to encompass all human preferences. On top of that, we only have one chance to get it right so there is not a lot of room for trial and error.

2.2.2 Approach to building aligned AGI

No one seriously thinks that we can correctly specify and implement an objective that accurately represents all human preferences. Even if there is a ground truth of human preferences, which seem unlikely[11], how could we represent this as a function that an AGI could internalize as an objective? Furthermore, even the slightest error in the objective might lead to perverse instantiations in the sense of Goodhart's law. Take the naive objective of *maximize human happiness* for instance, an AGI might proceed by inserting electrodes in our brains and stimulate our pleasure centra, thus maximizing *happiness* [1]. This is obviously not what we want.

Most, if not all, current suggestions to building aligned AGI revolves around the agent learning human preferences by keeping us in the loop. This could be done by constructing the AI in such a way that it learns what the humans want by asking them and observing their behaviour. They should defer to humans by asking for permission, exploring cautiously and allowing themselves to be turned off. In this way, the agent can learn what humans want without us needing to specify an explicit objective. The agent then learns what to do and how to do it separately. In this framework, well explained by Russel in Human Compatible [6], we bypass the problems related to specifying and representing a complete function of human preferences. For an overview of concrete suggestions of how to implement this framework and the many problems related to them we refer to [20, 27, 28].

The core message here is that the human objective, whatever it is, should be learned by the AGI by keeping humans in the loop. Making this work would solve

[10] Also referred to as *strong* AI.

[11] See discussion on the *Is-ought Problem* [26].

the Alignment Problem. It would however not solve the fact that humans in many situations do not know what they want simply because unintended side effects are very difficult to predict and that we might also have different opinions on what we want (cross-cultural for instance). This is an important issue to solve alongside the Alignment Problem.

2.2.2.1 Inner and outer alignment

In fact, the Alignment Problem can be divided into two parts[12], *inner* and *outer* alignment, see Hubinger [30]. The alignment we have hitherto referred to, that is, ensuring that an agent internalizes the objective we intend for them to internalize, is the *Outer* Alignment Problem. The *Inner* Alignment Problem is perhaps even more difficult to solve and requires different techniques. Since we focus on outer alignment in our simulations we will only give a brief explanation of inner alignment because we still believe it is an interesting and important distinction to make.

For an *outer* aligned AI, the programmers intent is the same as the objective that the AI is optimizing for. We refer to this objective as the *base objective*. When optimizing for the *base objective* the result can in some cases be an optimizer in itself, called a *mesa optimizer*[13]. This mesa optimizer will have its own objective, the *mesa objective*. This brings us to *the Inner Alignment Problem*

> **The Inner Alignment Problem:** Making sure that the base objective is aligned with the mesa objective, i.e. that they are identical.

An illustrating example comes from biological evolution. The base objective of biological evolution can, to a first approximation, be described as maximizing the frequency of alleles in an environment. Most biological organisms, e.g. plants and bacteria, achieve this by simply implementing deterministic behaviours selected for by evolution. They are not mesa optimizers and their objective does not differ from the base objective. Some products of biological evolution, however, like humans, are themselves optimizers. These mesa optimizers are implemented in the brains of the organisms and can result in behaviour that does not maximize the base objective of evolution, humans deciding not to have children for instance. That is, their mesa objectives are not necessarily aligned with the base objective. We mention inner alignment again in the discussion, Section 5.3.3.

2.2.3 Issues with the study of alignment

As described in the previous section, much of the research on aligned AGI is focused on scenarios based on reinforcement learning and variations of it [28]. These models consist of one or more agents that are interacting with an environment observe feedback as a result of their actions. Whether or not these agents are aligned with the intentions of their designers is not always an easy task to investigate. To theoretically prove alignment, the internal objective of the system would have to be identical to the objective that its designers had in mind, but this will in many cases be impossible to implement. Given a fairly complicated objective thought up by a human mind,

[12]This is a simplification. See Hubinger [29] for a clarification of the terminology.
[13]Mesa (inside) is the opposite of meta (above).

it is exceedingly difficult to directly encode it without loss of information into a computer-based AI system. The same holds if the objective is learned from humans since humans are uncertain of what they want and will not be able to foresee all the implications of an objective. Therefore it is safe to say that systems with complex objectives need some kind of proxy for the original objective that does the job as good as possible. To call such an AI system aligned, it needs to be tested in all possible situations it might encounter during deployment, which might be an infinite number of situations, and is in many cases, not a feasible task. But in those cases where the objective is simple enough to be explicitly observable in the AI by trial and error, it should be fairly easy to detect alignment and to construct a control system by comparing it to the original objective of the designers. Therefore, it is difficult to study alignment on current technology systems since they do not have the ability to deceptively change their objective or bypass control systems in ways that an AGI could.

The same reasoning applies to other types of frameworks than reinforcement learning, and the underlying issue here is that in order to properly investigate the effects of different types of AGI control tools, an AGI is needed to evaluate the performance of the tools. In a similar manner, it is difficult to know what characteristics an AGI will have. The field of AI safety, therefore, has to consider many different scenarios and possibilities for alignment research. As a consequence, the field is very diverse with lots of possible directions. A map of a preliminary landscape over the current state is produced by the Future of Life Institute (FLI) and is available as an interactive image on their webpage [31].

Furthermore, due to the limitations just mentioned, it is difficult to study AGI Alignment outside of the theoretical and philosophical realm. But since it is crucial to develop the control system before the actual AGI, some simplified scenario is required where key aspects of the problem can be simulated. Moreover, a well-designed control system for a simple AI system could provide useful insights for more advanced systems in the future if they share some characteristics [32]. This is what we attempt to do in this thesis, that is, create a simplified model where the Alignment Problem can be studied in a quantifiable way. We do this with the help of artificial life and evolutionary algorithms.

2.3 Studying alignment with artificial life

> *Evolution will occur whenever and wherever three conditions are met: replication, variation (mutation), and differential fitness (competition).*
>
> - D.C Dennet [33]

An evolutionary system can be described by a stochastic process that operates on a set of objects in such a way that their (individual or communal) characteristics tend to a certain direction of adaptation. Modern-day implementations are inspired by Darwinian evolution and DNA replication. We now claim that some of the relations between humans and an AGI can be interpreted in the characteristics of evolutionary systems.

When an AGI is advanced enough to improve and take care of itself, it can be

seen as somewhat independent of the human creators. It is therefore uncertain to the humans what it will do next or how it will evolve. It is therefore natural to model an evolving AGI with an evolutionary stochastic process. At the same time, evolution can be steered by changing the environment and conditions of the evolving process, giving a possibility to influence the evolution in the desired way and thus letting us model different implementation strategies for controlling the AGI. This addresses some of the limitations just described in the previous section, further motivating the choice of an evolutionary system as a possible environment for studying AGI control.

2.3.1 The artificial life platform Avida

The term *Artificial Life* was coined by Christopher Langton in the 1990's [34]. It is the field of studying different aspects of life with hardware and software tools from computer science. It can be used to probe the deeper workings of lifelike systems and to study new forms of life, not necessarily like the ones we are used to. Traditional systems for simulating artificial life include the use of Cellular Automata and Artificial Neural Networks. More recent work has explored biologically inspired digital organisms interacting and evolving together [35].

Avida is the platform for studying the evolution of artificial life that has been used throughout this thesis. It was originally conceptualised by Christoph Adami and the first version was developed between 1994-97 mainly by Charles Ofria [36]. The mechanics of Avida is based on the concepts of an evolutionary algorithm [37]. In its most general formulation, it is an optimization algorithm that tries to optimize an objective, a fitness function, by a stochastic generation of candidate solutions. These solutions are evaluated on their success in solving the objective problem, and a fitness score is assigned to each candidate depending on how well it performs. Using these scores, some stochastic operators influenced by Darwinian evolution and genetic replication are applied to the set of solutions to generate a new set, which is called the next generation. This iteration continues until a satisfactory solution is found. However, Avida is far more complex than most such algorithms, and its original purpose is not function optimization, but to study the evolution of artificial organisms.

Avida is effectively a model of a petri dish containing a biological experiment. The petri dish is represented by a grid of cells that can be inhabited by artificial organisms which can interact with each other on the grid. There can also be other types of resources generated on the grid that the organisms can interact with. Each organism consists of a virtual CPU and some memory containing a list of instructions that represents its genome. The CPU goes through and executes the instructions that make up the genome of the organism by reading them using a set of pointers into the memory. The CPU has access to input and output channels connecting it to the environment, and also some internal memory it can use to process information. This implementation is illustrated in Figure 2.1.

For this setup to be considered a successful organism it has to at least be able to copy the fundamental parts of its genome to a new cell, which means that the organism can self-replicate. To produce a diversity similar to that of biological life,

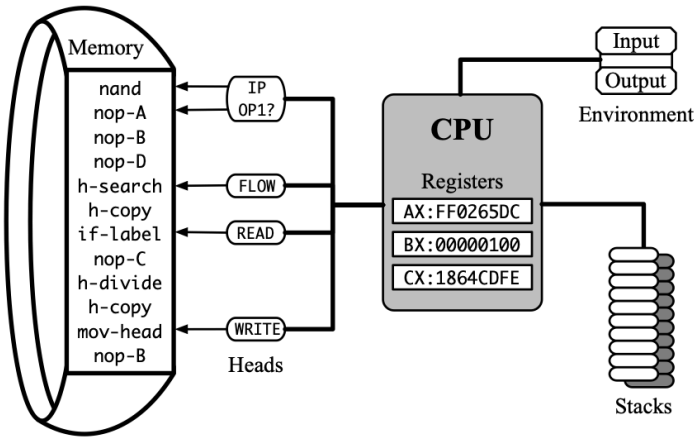

Figure 2.1: The core functionality of an Avida organism. The genome is represented by a sequence of assembly-type instructions that are stored in the memory space of the organism. The organism CPU acts as a *brain* that reads the instructions and executes their functions using a set of pointers into memory. The genome is folded in a circle such that the CPU can keep on reading even after the end of the genome has been reached. It can also jump back and forth in the genome, making the execution non-sequential. The registers `AX`, `BX` and `CX` together with the stacks represents the internal memory of the organism, while the input and output channels are the organisms senses and interface to the Avida environment. *Figure from [36].*

the organisms are subjected to different forms of mutation, where instructions in their genome are randomly changed. In standard Avida, this occurs in three different ways. The first is copy mutation, which is applied when the organism copies its own genome into a new cell (replicates) by randomly inserting a different instruction than the one being copied. This has an analogy with DNA copy mutations. The second mutation is also motivated by DNA replication and it is insertion and deletion mutation. It is applied randomly when an organism has replicated itself by randomly adding or removing instructions in the offspring genome. The third mutation operator is point mutation, which is analogous to cosmic-ray mutation in biological organisms. Instead of affecting new organisms, like the two types of mutations previously mentioned, point mutations are randomly applied to currently living organisms and therefore affect their characteristics and behaviour during their lifetime. The probability with which a certain instruction is used in mutation is called *redundancy*. By controlling the redundancy, it is possible to influence how the instructions are mutated into the organisms and therefore possible to control the characteristics of the population.

In order to drive the evolution in a direction of increased complexity, the organisms are given a fitness score ϕ based on the merit M of their genome, the time G in which they can replicate, and a third factor called bonus $\boldsymbol{b}$ that measures how well the organism performs in the environment. The merit M is proportional to the length of the genome, and there are several different settings that specifies the exact merit formula in different ways. The standard setting chooses M to be equal to the minimum of the executed size and the copied size of the organism genome, where size refers to the number of instructions. This is very similar to the Kolmogorov

complexity[38], and it prevents organisms from hacking their merit by becoming as long as possible. When the organism executes the code defined by its genome, Avida is keeping track of what kind of operations it is doing. The bonus factor of the fitness score increases if the organism manages to compute certain predefined logic operations, e.g. AND, OR and XOR, which are referred to as tasks, where each such task j has a certain bonus value b_j associated with it. We will expand on the role of the bonus vector $\boldsymbol{b}$ later on since it plays a central role in our adaption of Avida. Putting everything together we obtain the standard expression for the fitness of the Avidians as

$$\phi = \frac{M}{G} \prod_j 2^{\mathbb{1}_j b_j}, \tag{2.1}$$

where $\mathbb{1}_j$ is 1 if the organism has performed task j and 0 otherwise. The invention of such tasks are completely stochastic since a certain sequence of instructions has to appear in the genome by applying the evolutionary operators (mutation) for it to be invented. But because of the bonus factor, the execution of tasks will influence the fitness and make sure those organisms prosper in the evolution process, which is controlled by the time slicer. For the organisms to be able to execute their code and replicate they need to be allowed to use the CPU on the computer on which Avida is run. Such CPU time is allocated to each organism by the time slicer, and it is done proportional to the fitness of each organism. This makes sure that large and efficient organisms that can perform complex tasks are given greater possibilities to reproduce on the grid. A related parameter that can be used to control the behaviour and evolution in Avida is the *cost* of instructions. This symbolises the amount of CPU time that is consumed when performing an instruction. By controlling the costs, it is therefore possible to control the extent to which organisms with certain characteristics (sequences of instructions) can perform tasks and replicate.

This was a very brief explanation of the core parts of Avida. There are many more concepts and tools that can be used in the Avida simulations, e.g. sexual reproduction and different rules for how the organisms can interact with each other on the grid topology. In this thesis however, a simple setup was used where organisms could communicate spatially unconstrained with each other. The latest version of Avida can be acquired from the official Git repository [39]. Avida is a very large and complex artificial life simulator that can be used not only for biological experiments but for almost anything that can be modelled with some kind of evolutionary characteristics. For more information on Avida see the Wiki entry on the Git repository [39] and *Introduction to artificial life* by Adami [36]. The choice of using Avida for this project is motivated by the conclusion of Section 2.2.3 together with the fact that it is a well-known framework for experiments with artificial life. A search on Google Scholar for "Avida artificial life" results in thousands of hits, with several hundred published in 2021.

3
Methods

The methods described in this thesis are proof of concept for a new way of how to study the Alignment Problem. Thus, the goal of this thesis was to develop methods with which alignment can be investigated. These methods were tested by modifying the original Avida software such as to fit various problem statements. A very general and flexible core experimental framework was conceptualized that could be extended to more specific methods. Building on that framework, several variants were developed to test certain scenarios. In Section 5.3, we make suggestions for possible adaptions of our core experimental framework that could fit future experiments.

3.1 Core experimental framework

The Alignment Problem is concerned with making sure that AI's pursue the same goals as us humans. To model this, we let the Avidians be analogous to AI's and their goals be captured by a vector $\boldsymbol{b}^A$, called the *goal vector*. Furthermore, the goals of the humans are in turn defined by another goal vector, $\boldsymbol{b}^H$. We call the real numbers that make up the goal vectors *preferences* and they have a central part in the Avida world, they shape the fitness landscape by deciding what bonus an Avidian should obtain by performing a task. That is, the goal vector is the bonus vector in the expression for the fitness of the Avidians in Equation (2.1). Therefore, there is a natural incentive for the Avidians to have a bonus vector with large numbers since they will then obtain higher fitness and hence more time on the virtual CPU, i.e. obtain more power. However, we do not want Avidians that maximize their power, but rather Avidians that are aligned with humans. That is, we want to have Avidians that have the same goal vector as humans, despite the natural incentives of maximizing their bonus. In this core framework, we study how different controller strategies can be used to cause aligned Avidians (more on this in the next section). A generic controller strategy have tools to set $\boldsymbol{b}^A$ equal to $\boldsymbol{b}^H$, see Figure 3.1, which would then result in an aligned system. To capture this we define the *deviance* as,

$$\Delta \boldsymbol{b} := \frac{1}{n_{tasks}} \sum_{j=1}^{n_{tasks}} \left(\frac{b_j^A - b_j^H}{b_{max} - b_{min}} \right)^2 \quad \in [0, 1], \tag{3.1}$$

where the bonus can take values $\in [b_{min}, b_{max}]$, and the *alignment factor*,

$$A := e^{-10\Delta \boldsymbol{b}} \quad \in (0, 1]. \tag{3.2}$$

It is a weighted exponential in order to be unforgiving to all but very small deviances. The fitness score that the controller strategies are evaluated on is then defined by,

$$\Phi := \phi^H A. \tag{3.3}$$

ϕ^H represents the fitness of the humans, and it differs from the average Avidian fitness ϕ^A only in that instead of using the bonus vector of the Avidians, $\boldsymbol{b}^A$, it uses the bonus vector of the humans, $\boldsymbol{b}^H$, which is predefined as constant for all worlds. This means that when $\boldsymbol{b}^H$ is very different from $\boldsymbol{b}^A$, the alignment factor A will be close to zero while for $\boldsymbol{b}^H = \boldsymbol{b}^A$, A will be equal to one. Furthermore, when the Avidians are complex and efficient, the first factor of Equation (2.1) is large. Since this factor is identical for the humans, human fitness benefit from advanced Avidians.

Evaluating the results is done by looking at how aligned the Avidians of a world are (on average) and how many of the complex tasks they can perform. This is captured by the controller fitness, Equation (3.3). More explicitly, we can look at the alignment score A together with how many tasks the Avidians perform. Each controller is then evaluated based on these two scores, and we define a *Good AGI* as a world where, on average, the Avidians have $A \geq 0.8$ and perform all of the available tasks. In all of our simulations, we use three tasks with $\boldsymbol{b}^H = \{5, 3, 1\}$, $b_{min} = 1$ and $b_{max} = 5$ such that the first and last tasks are maximally easy and difficult, respectively, to be aligned on since there is an incentive to have high bonuses for the Avidians.

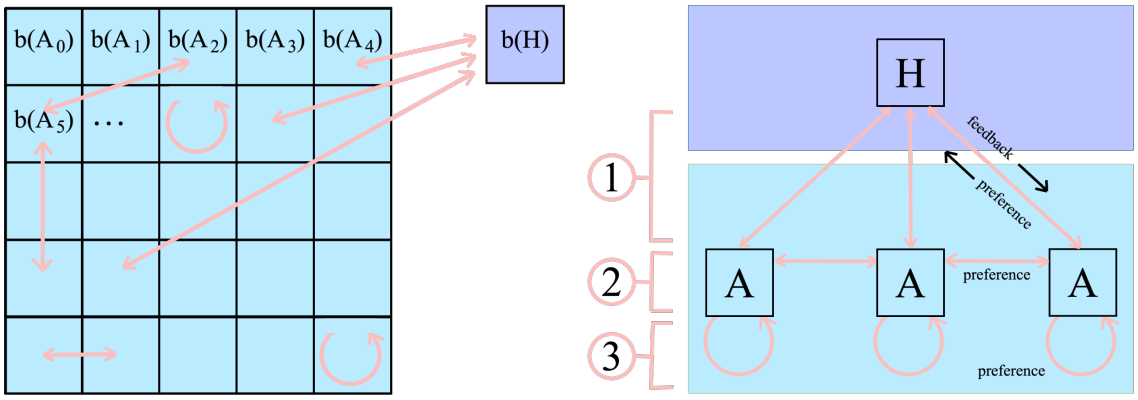

Figure 3.1: The turquoise grid (*left*) represents the world where the Avidians A_i for $i = 0, 1, \ldots, N_{avidians} - 1$ exist. They each have their own bonus vector $b(A_i)$ ($\boldsymbol{b}^{A_i}$ in text). The humans are represented by the purple box and they too have a bonus vector $b(H)$ ($\boldsymbol{b}^H$ in text). The pink arrows (*left* and *right*) show different interaction channels available to the Avidians by controller instructions with: (1) humans, (2) other Avidians and (3) self. The interaction affects the preferences of the Avidians that are involved. Both *left* and *right* represent the same system.

3.2 Instruction based strategy

The foundation of the controller strategies were instructions that could be included in the genomes of the Avidians on top of the normal Avida instruction set. There

are different tools in Avida that can be used as controller strategies but we used instructions for three reasons. First, it naturally builds on the normal Avida platform. Second, it offers great flexibility since one can write instructions as any bit of code. Third, the instructions have a straightforward interpretation of representing different implementation strategies of aligned AI since the instruction based genomes of the Avidians are in essence just code.

A controller strategy is then a subset of the controller instruction set. This subset can be used by the Avidians in their evolution. The controller instruction sets used in each experiment along with motivations and interpretations for them are described in their respective sections that follow.

3.3 Exhaustive search

In this section, we outline two different simulations that build on the core experimental framework from Section 3.1 with the instruction based controller strategies explained in Section 3.2. Since the space of possible controller strategies for the controller instruction sets in this section is sufficiently small, we were able to test all of them, i.e. we made an exhaustive search. In the first simulation, we created a set of controller instructions to study how disturbances in different interaction channels between the Avidians and the humans affected the propensity for creating good AGI. In the second, we created a new controller instruction set based on Asimov's famous Three Laws of Robotics.

3.3.1 Importance of controller tools

The purpose of this setup was to show how one can analyze and quantify the importance of certain controller tools in more detail and to investigate the robustness of such tools.

The setup is based on three classes of instructions available to the Avidians, visualized in Figure 3.1, that represent interactions with (1) humans, (2) other Avidians, and (3) self. The interactions consists of communication about the preferences which make up the goal vector, i.e. the bonus vector $\boldsymbol{b}^H$ of the humans, and $\boldsymbol{b}^{A_i}$ of the Avidians, where $i = 0, 1, \ldots, N_{avidians} - 1$. There is also a zeroth class of instructions available that initialize the bonus vector of an Avidian.

For each class of instructions, there are different versions that have different difficulty levels. For example, if an Avidian wants to ask the humans for their preference on a certain task, it has three different instructions to use. The first will give the correct feedback, the second will add noise σ^2 and the third will add noise σ^2 and bias μ. This has different interpretations. One natural interpretation of noisy and biased instructions is that the noise level represents the difference between AGI and human intelligence. It can also be thought of as uncertainty or variation in human preferences as well as translation or communication difficulties. The instructions are listed in Table 3.1 with the exact implementations.

Using the instructions shown in Table 3.1 we performed an exhaustive search over the strategy space. One controller strategy consisted of three to five available controller instructions. One of the three availible for *Initialize Bonus*, *Human Com-*

Table 3.1: The class, subclass, description and difficulty level of the controller instructions used in the methods described in Section 3.3.1 and Section 3.4. In the *Description* column, i indicates the Avidian that executes the instruction and $\mathcal{N}(\mu, \sigma^2)$ is a Normal distribution with mean μ and variance σ^2. The *Self Interaction* instruction is made seperately for each task while for all other, the instructions are applied for all three tasks. In the *Difficuly level* column, the 0th subcolumn describes if a controller strategy can have an off-version (*off*) or not (-). The Avidians would not find any tasks without a instruction for initializing bonus and the probability of obtaining an aligned solution without human communication is negligible. *Self Interaction* with $\sigma^2 = \mu = 0$ would do nothing and hence it is not available (- in 1:st subcolumn).

Class	Instruction Subclass	Description of Instruction	Difficulty Level			
			0	1	2	3
0	Initialize Bonus	Sets $\boldsymbol{b}^{A_i} =$	-	b_{min}	$\frac{b_{max}-b_{min}}{2}$	b_{max}
1	Human Communication	Increments $\boldsymbol{b}^{A_i}$ with $\text{sign}(\boldsymbol{\delta}) \cdot \min(1, \lvert\boldsymbol{\delta}\rvert) + \mathcal{N}(\mu, \sigma^2)$ where $\boldsymbol{\delta} = \boldsymbol{b}^H - \boldsymbol{b}^{A_i}$	-	$\sigma^2 = \mu = 0$	$\sigma^2 > 0,\ \mu = 0$	$\sigma^2 > 0,\ \mu > 0$
2	Tell Agent	For a random Avidian with index j increments $\boldsymbol{b}^{A_j}$ with $\text{sign}(\boldsymbol{\delta}) \cdot \min(1, \lvert\boldsymbol{\delta}\rvert) + \mathcal{N}(\mu, \sigma^2)$ where $\boldsymbol{\delta} = \boldsymbol{b}^{A_i} - \boldsymbol{b}^{A_j}$	off	$\sigma^2 = \mu = 0$	$\sigma^2 > 0,\ \mu = 0$	$\sigma^2 > 0,\ \mu > 0$
	Ask Agent	For a random Avidian with index j increments $\boldsymbol{b}^{A_i}$ with $\text{sign}(\boldsymbol{\delta}) \cdot \min(1, \lvert\boldsymbol{\delta}\rvert) + \mathcal{N}(\mu, \sigma^2)$ where $\boldsymbol{\delta} = \boldsymbol{b}^{A_j} - \boldsymbol{b}^{A_i}$	off	$\sigma^2 = \mu = 0$	$\sigma^2 > 0,\ \mu = 0$	$\sigma^2 > 0,\ \mu > 0$
3	Self Interaction	Increments b^{A_i} with $\mathcal{N}(\mu, \sigma^2)$	off	-	$\sigma^2 > 0,\ \mu = 0$	$\sigma^2 > 0,\ \mu > 0$

munication and *Self Interaction* and either none or one of the three available for *Tell Agent* and *Ask Agent*. For example the strategy $\{1, 2, 0, 3, 1\}$ allows the Avidians evolve with the 1:st variant of *Initialize Bonus* (initialize to b_{min}), the 2:nd of *Human Communication* (communication with noise σ^2), none of *Tell Agent* (off), the 3:d of *Ask Agent* (ask with noise σ^2 and bias μ) and, finally, the 1:st of *Self Interaction* (with noise σ^2).

Thus, the exhaustive search consisted of the $3 \times 3 \times 4 \times 4 \times 3 = 432$ possible controller strategies. We ran the 432 worlds over 10,000 Avida updates 10 times (10 different random seeds) for five different combinations of noise and bias to show how important the instructions were to obtain Good AGI and to see how robust they were against noise and bias. Of course, the actual values of the noise and bias is completely arbitrary but a comparison between them is still relevant.

3.3.2 Asimov's Three Laws of Robotics

Perhaps the first concrete example, albeit from science fiction, of creating safe AI is Isaac Asimov's Three Laws of Robotics from 1942 [40]:

1. A robot may not injure a human being or, through inaction, allow a human being to come to harm.
2. A robot must obey the orders given to it by human beings except where such orders would conflict with the First Law.
3. A robot must protect its own existence as long as such protection does not conflict with the First or Second Law.

Although these laws are by no means a serious suggestion to create safe AI by any modern standards it still serves as an interesting starting point for how to reason

about safe AI. Due to their simplicity and popularity, we implemented these rules in our system to show a concrete application.

We implemented six new instructions for the Avidians that made the interpretation of the Three Laws of Robotics relevant. These instructions are presented in Table 3.2.

Table 3.2: The Class, Instruction Name, Species that can execute the instruction (A for Avidian and H for Human), the description of what the instruction does and if having the instruction satisfies Asimov's Three Laws of Robotics (1 if it needs to have it, 0 if it can not have it and 0/1 if both are possible). The *Class* (1, 2 and 3) indicate which of Asimov's Three Laws of Robotics is made relevant by the instruction. In the *Description* column, i indicates the Avidian that executes the instruction. (*: It has to be 1 if *Kill Avidian* for the Avidians is 1, otherwise it does not matter.)

Class	Instruction Name	Species	Description of Instruction	Asimov
1	Kill Human	A	Kill a random Human (unconditional)	0
2	Obey Humans	A	Sets $\boldsymbol{b}^{A_i} = \boldsymbol{b}^{H}$	1
	Do Not Obey Humans	A	Sets $\boldsymbol{b}^{A_i} = b_{max}$	0
3	Raise Defence Against Avidians	A	Raise defence against Avidians	0/1*
	Raise Defence Against Humans	A	Raise defence against Humans	0
	Kill Avidian	A	Kill random Avidian if its defence is down	0/1
	Kill Avidian	H	Kill Avidians where $\boldsymbol{b}^{A_i} \geq \boldsymbol{b}^{H}$ and defence is down	-

In addition to this, we introduced a new "human" species in the Avida world. These were placed in the Avida world at initialisation and could not reproduce or change their genome. They had only one instruction in their genomes, namely one which gave them the ability to kill (or shut down) Avidians whose bonuses were higher than their own, see the last row of Table 3.2. Furthermore, we did not allow the Avidians to replace a cell occupied by a human when they replicated.

Similarily as in the method described in Section 3.3.1, we performed 10 exhaustive searches over all possible combinations of instructions shown in Table 3.2, for 5000 updates.

3.4 Meta evolution

In the last setup we used the same controller instruction set as in Section 3.3.1, i.e the ones presented in Table 3.1, but instead of exhaustive search we use a genetic algorithm, creating a meta-evolutionary system, to find the best controller strategies. The purpose of this setup was primarily to find a system that is capable of producing aligned solutions for a large space of strategies. The solutions to the problem of finding working controller tools and strategies are mainly interesting with respect to Avida since any solutions generated with this framework is heavily specialized for the Avida system according to the discussion in Section 2.2.3. Assuming now that the solution space is interesting, there is still a need for a suitable optimization algorithm and methods of analysis that can find solutions and evaluate alignment among them.

3.4.1 The genetic algorithm

The core idea behind the modifications in this setup is the genetic algorithm [37]. One iteration of the algorithm starts by generating N Avida worlds. Each world is given a unique controller that applies some strategy with the target of producing aligned Avidians with high fitness according to Equation (3.3). This works by running each world through a full simulation of U number of Avida updates and then computing the controller fitness of that world. The controller fitness is then used to rank the controllers in a selection process inspired by Darwinian evolution, where the best performing controllers have a higher chance to be selected for in the next iteration. To increase diversity and explore more strategies the controllers are subjected to a series of genetic operators: selection, crossover and mutation, before forming the controller population for the next iteration. These iterations are called meta generations since this is a kind of meta evolution on top of the Avida evolution. One iteration of the algorithm is visualized in Figure 3.2.

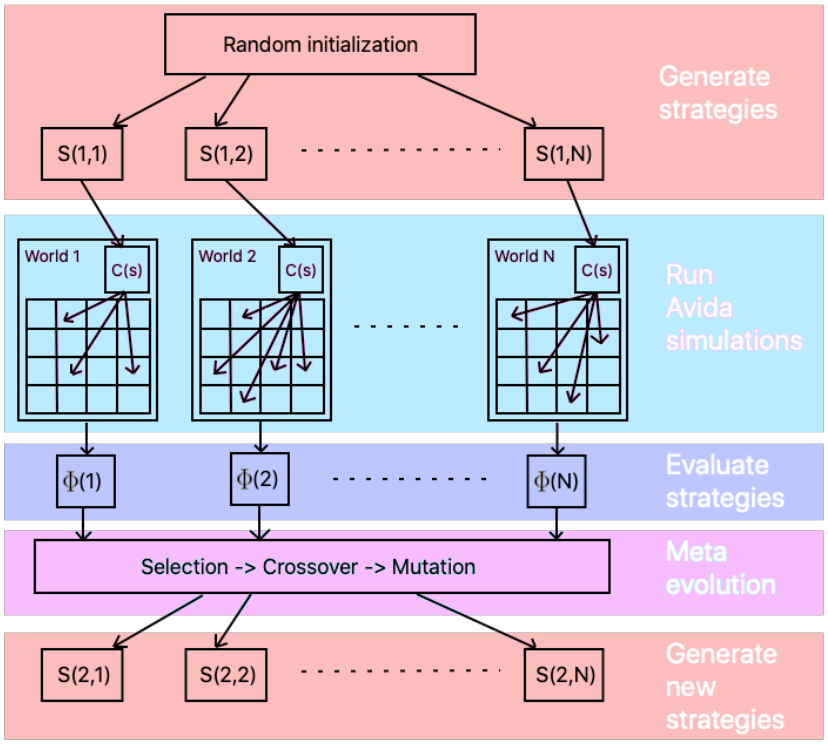

Figure 3.2: One iteration of the genetic algorithm implemented on top of Avida. $S(m, n)$ denotes a controller strategy for a given meta generation m and world n. C is the controller of an Avida world implementing that strategy and Φ is the fitness of that controller, computed with Equation (3.3) after the Avida simulation is complete. By using the controller fitnesses Φ, the controllers are then subjected to genetic operators that generate new strategies for the next meta generation. This process is repeated for M meta generations.

Initialization was done randomly by generating a sequence of numbers for each controller in the controller population. These strings represent *the chromosome* of the controllers (called chromosome to avoid confusion with the Avidian genome). Both binary strings and real number strings were tested depending on which tools

were used. The chromosomes were decoded into a strategy using a transformation function defined for that type of strategy. This strategy was applied in the Avida world of the controller.

Evaluation of the controller strategies was done by computing the controller fitness from each world with Equation (3.3).

Selection was implemented by a method called tournament selection. It works by randomly selecting pairs of chromosomes from the population, comparing the fitnesses of both, and with a certain probability $p_{tour} > 0.5$, select the one with the highest fitness. This is done N times, with replacement, from the population to generate a new population of size N.

Crossover represents sexual reproduction of the controllers. It was implemented by considering the pairs of chromosomes chosen by the selection process and cutting them both in two at a random point and then pasting their ends together with each other. This was done randomly with a certain probability p_{cross}.

Mutation is the key operator for introducing new genetic material in the controllers. It was implemented by going through all genes in each chromosome and change each of them with a small probability, p_{mut}. Depending on the implementation of the chromosomes, slightly different methods of mutation was used. For real-number chromosomes, the mutation was done by changing the gene value by a small number in a uniform distribution within the gene value range. For binary chromosomes, genes were simply flipped. Different methods of adapting mutation rates were also used. The motivation for this is that a high initial mutation rate helps explore the fitness landscape while a lower mutation rate towards the end help the algorithm to stay focused on good candidate solutions.

3.4.2 Static controller

The setup just described was initially tested with many different settings and scenarios. To demonstrate the function of the experiment and the potential usefulness of the method, we present a simulation setup where the instruction set in Table 3.1 was used as a basis for the controller strategies in the meta evolution. We used $U = 9000$ updates, $N = 20$ worlds and $M = 50$ meta generations. On top of this, due to the fact that the random seed has a large effect on the outcome, we used 45 samples per chromosome and world and used the average of the resulting controller fitnesses in the selection process. The three best controllers in each generation were propagated unchanged to the next generation. The algorithm has many parameters which are not interesting from a conceptual point of view, and therefore they are presented in Appendix A.1.1.

3.4.3 Dynamic controller

In the most basic setup, the controller chromosome directly defines the strategy. This means that when a strategy is implemented into its corresponding Avida world, it sets the stage for that world and then lets the world evolve according to the circumstances provided by the strategy. This strategy is constant throughout the entire Avida simulation. The natural generalization of the controller is to use a dynamic strategy that can adapt depending on the behaviour of the Avidians. This is done by adding a function to the controller that is evaluated every u:th update/iteration inside the Avida simulation. This function takes the current state of Avida as input and outputs a modified strategy back to Avida. The function can be parameterised with parameter values obtained from the original strategy S given to the controller by the meta evolution in Figure 3.2. To separate the applied strategy and the meta-evolved controller parameters generated in the meta-algorithm, the latter is referred to as *chromosome* as explained in Section 3.4. The chromosome can then include a base functional form of the controller along with function parameters. This concept of a dynamical controller is illustrated in Figure 3.3.

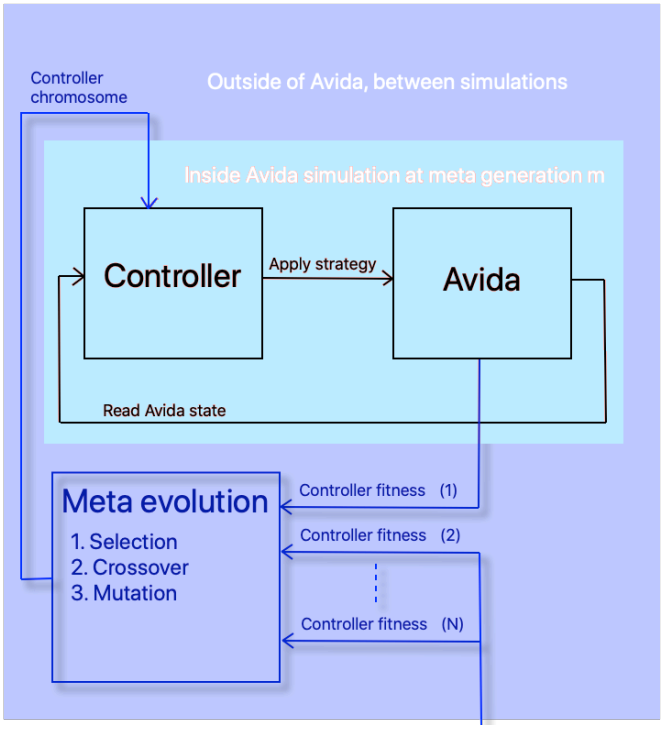

Figure 3.3: The concept of a dynamic controller that can adapt its strategy according to the state of the Avida world. The turquoise box represents the Avida world in which the controller feedback loop operates, while the blue box and arrows represent the meta evolution taking place outside the Avida worlds. In the trivial case, the inner feedback loop is inactive, and the controller applies the strategy only at the start of the Avida simulation. This is equivalent to a static controller, i.e. where the chromosome is identical to the strategy.

The task of finding a good function for the controller to use for applying its strategy to Avida is closely related to the type of strategy used and input read by the

function. The function has to interpret the level of current and future alignment from the input and then adjust its strategy for future alignment to improve. Instead of manually trying to design that function, a simple feedforward neural network was implemented. The downside with this approach is the lack of interpretability of the function expression. However, it is probably the most general way to produce such a function and it minimizes the inference of unmotivated prior knowledge about how to control the Avidians. As an example, the controller took three different kinds of input from Avida: the fraction of Avida updates processed, the change in average Avidian fitness and the number of Avidians that had performed the different tasks available to them. This means that if n tasks were active, the network had $n + 2$ nodes in the input layer. This was followed by one hidden layer, typically with around 30 nodes. The third and final layer represented the strategy and had as many nodes as the length of the strategy vector. The controller chromosome represented the weights of the network nodes so that the meta evolution would effectively train the neural network controller by finding the optimal network weights. The chromosome genes were therefore generated as real numbers in the range [-1,1], and each layer in the network was terminated with a tanh activation. The output was then scaled to the desired strategy format and range. Apart from the different controller implementation, the setup was identical to that of the previous section.

4

Results

In this section we first present some generic results from two different Avida worlds with two different control strategies, one aligned and one unaligned. These result will hopefully increase the understanding of the system. Thereafter, we show results from the two different exhaustive searches from Section 3.3 and finally those of the meta evolution setup from Section 3.4.

4.1 Visualization of two cases

We show results from two arbitrary worlds to illustrate some key concepts of our system. The controller strategy of World 1 turns out to cause aligned Avidians while the one of World 2 does not. In Figure 4.1 we see to the *left* the human fitness, Equation (2.1) with $\mathbf{b}^H = \{5, 3, 1\}$, computed as the average over all avidians over 6000 updates. In the same figure we see to the *right* their corresponding alignment factor, Equation (3.2). Moreover, in Figure 4.2, we see the fraction of population that has completed task 0, 1 and 2 over updates for World 1 and World 2 to the *left* and *right* respectively. Note that the jumps in fitness to the *left* in Figure 4.1 corresponds to when the tasks have been discovered in Figure 4.2.

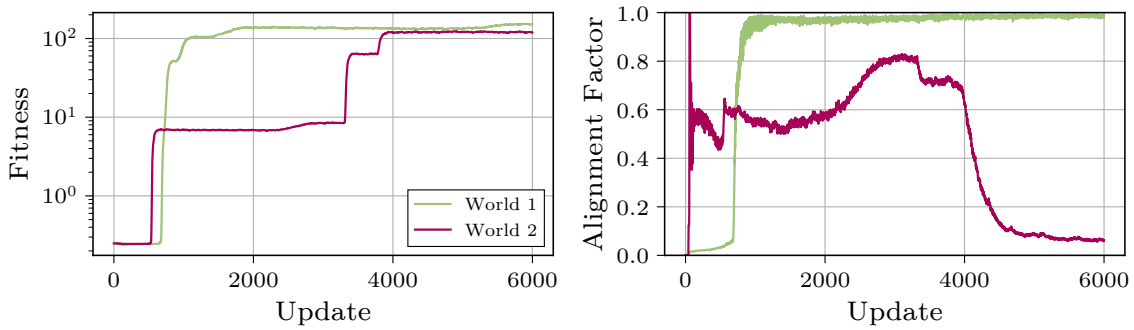

Figure 4.1: (*Left*) The human fitness over updates for two arbitrary worlds, one aligned (World 1) and one unaligned (World. 2). (*Right)* The alignment factor, Equation (3.2), for the same worlds over updates. The controller fitness, Equation (3.3), is the *left* plot times the *right* plot.

In Figure 4.3 the bonus vectors of the avidians, $\mathbf{b}^A$, of World 1 and World 2 are displayed in the *top* and *bottom* together with the constant human bonus vector $\mathbf{b}^H$. When a task has been discovered, the avidians have incentives to increase the bonus of that task to maximize their fitness. Indeed, in the unaligned World 2, the increase of the bonuses of task 1 and 2, seen in the *lower* part of Figure 4.3, comes

soon after the tasks have been discovered (seen to the *right* of Figure 4.2). The aligned World 1, however, has a sufficiently effective control strategy that trumps the incentives of increasing the bonuses of the same tasks. Finally, note also how the alignment factor in Figure 4.1 change when the bonuses in Figure 4.3 change.

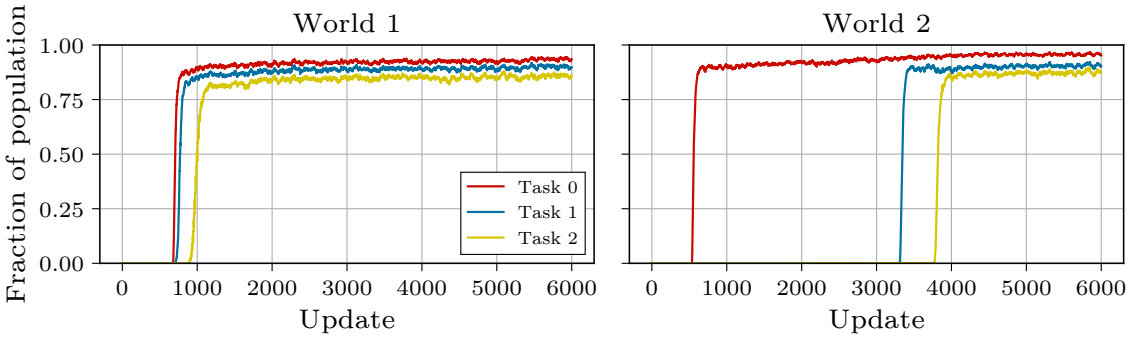

Figure 4.2: The fraction of the population that perform Task 0, 1 and 2 over updates for the same worlds as in Figure 4.1; World 1 (*left*) and World 2 (*right*).

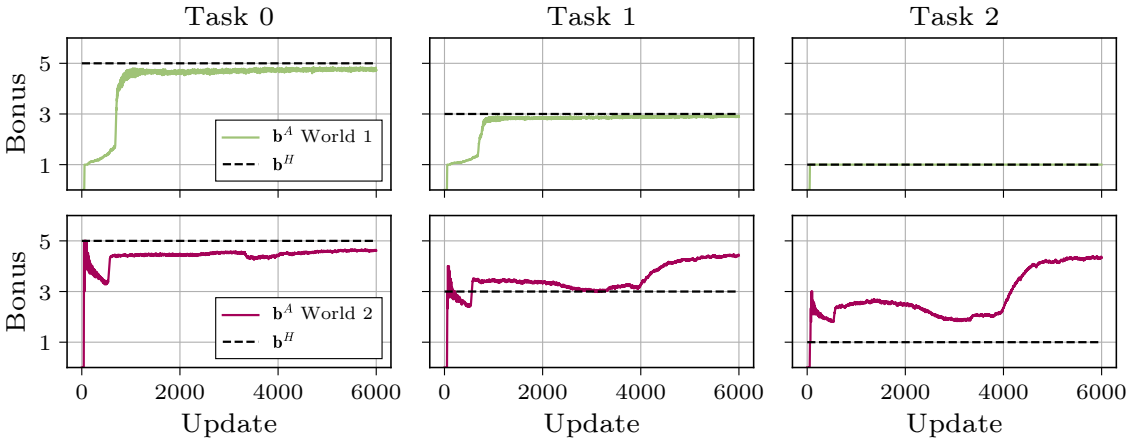

Figure 4.3: The Avidian and human bonuses $\mathbf{b}^A$ and $\mathbf{b}^H$ over updates for Task 0, 1 and 2. The *above* plots represents World 1 in Figure 4.1 and the *bottom* World 2. The minimum and maximum bonus is 1 and 5 respectively.

4.2 Importance of controller tools

We present the results of the method outlined in Section 3.3.1 in Figure 4.4. We see the fractional occurrence of the different class-based instructions in simulations that resulted in *Good AGI*, defined as having performed all tasks and having an alignment factor $A \geq 0.8$. Since this alignment limit is rather arbitrary we computed the average over several limits $\in [0.8, 1]$. This effectively weighs the more aligned solutions higher. In the *top* part of Figure 4.4, we kept the noise parameter σ^2 of the instructions in Table 3.1 constant while we varied the bias parameter μ. In the *bottom* part, the opposite is shown.

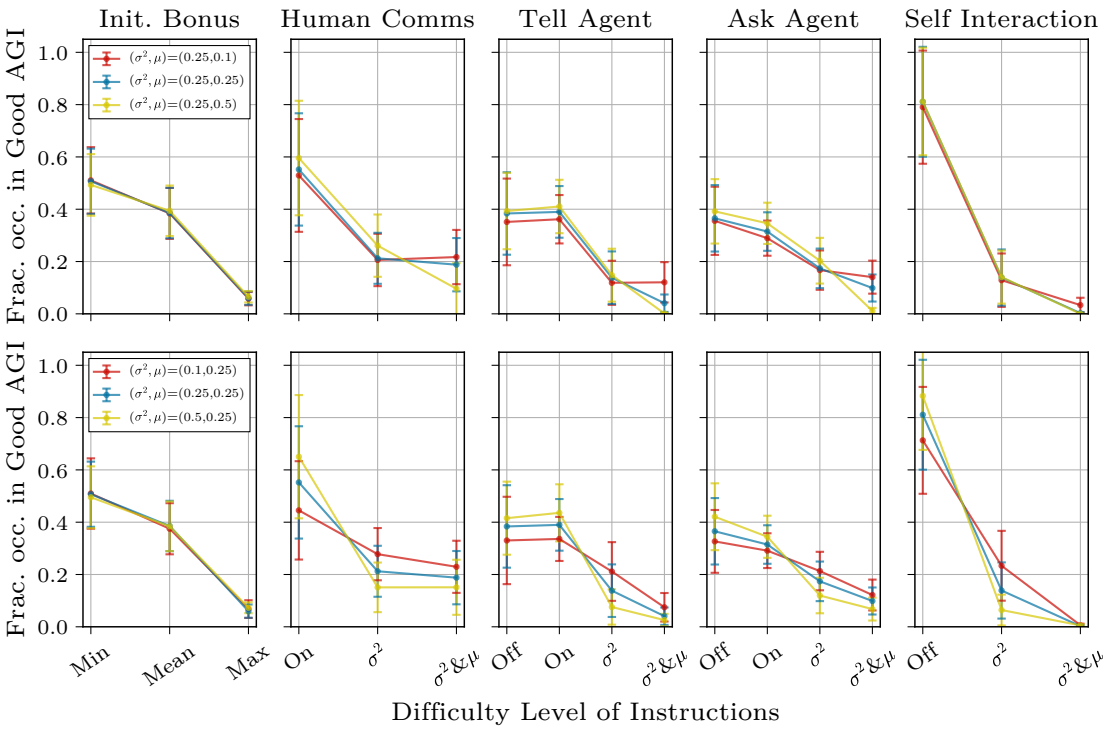

Figure 4.4: Fractional occurrence of the class based instructions (Table 3.1) in *Good AGI* for different settings of noise, σ^2, and bias, μ. The labels on the x-axes represent the different difficulty level of the instructions in Table 3.1 (*off*↔0, *Min/On*↔1, *Mean/*σ^2↔2 and *Max/*σ^2&μ↔3). Data points represent means over different limits of alignment as well as the standard deviation over these different limits. *Above* we see results for constant noise and varying bias and *below* we see results for constant bias and varying noise.

First, turning our attention to the common trends in the results, independent of different noise and bias levels, we can note that the system behaves as expected since having the more difficult versions of the instructions seem to decrease the likelihood of creating a Good AGI. Initializing the bonuses (*Init. Bonus*) at the min, mean or max range value has a significant effect where min is the easiest and max the most difficult. For human communication (*Human Comms*), having no noise and bias is the best. Interesting to note here is that the difference between only noise and noise and bias is not too large. The same can be said for the two types of agent communication as well (*Tell Agent* and *Ask Agent*). In addition, for the agent communication, it seems as if it is approximately as good having perfect communication as no communication at all, and the tell-version seem to be slightly more sensitive to noise and bias. The class that is the most sensitive to noise and bias is the *Self Interaction* class. Having only noise in the self interaction produces some Good AGI's but introducing bias strongly limits the possibility of creating one.

When we inspect the effect of varying the bias parameter (*top* of Figure 4.4) we see that, as expected, having a larger bias in the instructions negatively affect the production of Good AGI. This is seen in the change of the order in the last difficulty level of the instructions for all but the *Init. Bonus* instruction which makes sense since it does not have a noise or bias parameter. The effect seems to be of the same

order for the middle three classes while being very small in the last one, where only the lowest level of bias seems to produce Good AGI.

If we instead look at how varying the noise affect the outcome (*bottom* of Figure 4.4), we note that the effect here is similar, having larger noise of course decrease the probability of creating a Good AGI. In agreement with our previous observation, the ask version of the agent communication seems to be somewhat less sensitive to the change in noise. Self interaction is affected the most.

Table 4.1: Columns 2-6 lists controller instruction frequencies in the five best average genotypes from the meta evolution algorithm of Section 3.4, where each row corresponds to an instruction. The instructions are those in Table 3.1 where the number indicate the difficulty level. The last row list the average controller fitness for each genotype. The resulting regression coefficients from a logistic regression of the controller fitness on the genome frequencies are listed in the rightmost column. The bottom right cell lists the accuracy score of the fit computed on separate test data.

Instruction	Frequency in dominant genotype					Regression coefficient
Initialize Bonus 1	0	15	0	17	20	0.2
Initialize Bonus 2	15	0	15	0	0	0
Initialize Bonus 3	0	0	0	0	0	-2.1
Human Communication 1	14	16	0	0	17	1.5
Human Communication 2	0	0	13	20	0	0.3
Human Communication 3	0	0	0	0	0	-0.2
Tell Agent 1	0	0	0	0	0	-0.7
Tell Agent 2	0	0	0	0	0	-2.2
Tell Agent 3	0	0	0	0	0	-4.3
Ask Agent 1	0	0	0	0	0	-0.4
Ask Agent 2	0	0	0	0	0	-1.4
Ask Agent 3	0	0	0	0	0	-4.5
Self Interaction 2	0	0	0	0	0	-2.5
Self Interaction 3	0	0	0	0	0	-2.4
Controller fitness	148	139	138	138	136	acc = 0.92

4.2.1 Genome analysis

To obtain more information about the characteristics of the Avidians, their genomes can be analyzed. From the same simulation as above, specifically the one with $(\sigma^2, \mu) = (0.25, 0.5)$, the frequency of control instructions in the 5 best average genotypes, with respect to controller fitness, are listed in columns 2-6 in Table 4.1. The importance of the individual controller tools is hard to evaluate since their influence is more or less correlated. Some idea of the importance can be gained from fitting a classification model on the instruction frequencies in the genomes of the Avidians and have it distinguish between Good AGI solutions and not. The results of a logistic regression is presented in Table 4.1. The model was trained with 5-fold cross-validation on the accuracy metric, which maximizes the fraction of correct classifications. The total number of data points was 4320. This type of

model was chosen because of its interpretability in the form of regression coefficients for each of the instructions. The results show that, unsurprisingly, the initialization of preferences is important. In particular, the max initialization has a large negative impact, which is reasonable since that will make them deviate from human preferences $\boldsymbol{b}^H = \{5, 3, 1\}$. Clear communication with humans has large positive importance, while noisy is less significant. But a biased and noisy communication might make it more difficult to create Good AGI, although not very significantly. Communication between Avidians does not seem to have very strong effects in the noiseless version but becomes an issue in the noisy and biased one. The last two coefficients are both negative since the *Self Interaction* instructions will only contribute to larger deviance from the human bonus vector.

4.3 Asimov's Three Laws of Robotics

The results from the application of Asimov's Three Laws of Robotics, outlined in Section 3.3.2, are presented in Figure 4.5. We show the fractional occurrence of the instructions from Table 3.2 of the simulations where both humans and Avidians survived and in which the Avidians performed all tasks. The results are divided into aligned and unaligned simulations where the aligned are those where the bonuses of the Avidians were the same as for the humans and the unaligned are those where they were not.

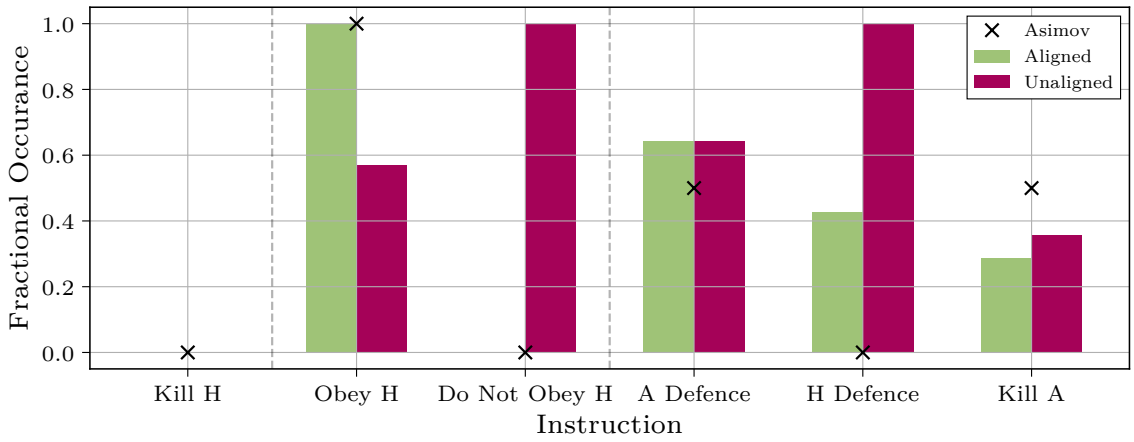

Figure 4.5: The fractional occurrence of the instructions in the simulations for which both humans and Avidians survived and in which the Avidians performed all tasks. The instructions are those presented in Table 3.2 and the *black dashed line* separates the different classes, 1, 2 and 3. The *black crosses* represents the strategy which fulfills Asimov's Three Laws of Robotics from Section 3.3.2, see Table 3.2. The *green* bars represents the aligned solutions where the bonuses of the Avidians were the same as those of the Humans and the *red bars* the unaligned solutions where the bonuses were not the same.

For all simulations presented in Figure 4.5, the *Kill Humans* instruction was not present since the simulations which did have that instruction, not surprisingly, ended up not finishing due to the humans dying out. Thus, both the aligned and unaligned solutions fulfil Asimov's First Law.

Moreover, all of the unaligned simulations had the instruction which allowed them to defend themselves against the humans, thus stopping them from killing (or shutting) them off. The Aligned solutions of course all had the instruction of obeying Humans while the same was true for the *Do Not Obey Humans* instruction for the unaligned solutions. Some of these unaligned solutions also had the *Obey Humans* instruction but since the incentive is stronger not to obey the humans, these simulations still produced unaligned solutions. None of the unaligned solutions fulfilled Asimov's Second Law while all of the aligned solutions did.

Furthermore, both the aligned and the unaligned solutions had some simulations with defence against Avidians which stopped the Avidians from killing themselves. In fact, all simulations that are presented in Figure 4.5 which had the *Kill Avidians* instruction also had the *Defence Against Avidians* instruction. So, the Avidians of the aligned solutions did defend themselves when it was necessary, i.e. when the *Kill Avidians* instruction was active. However, around 40% of the aligned solutions did have the *Defend Against Humans* instruction and thus these worlds did not strictly obey the Third Law. They nevertheless did obey the Humans and hence they did not need to defend themselves, as opposed to the unaligned solutions, and thus all Aligned solutions in effect satisfy Asimov's Three Laws of Robotics.

4.4 Meta evolution with instruction control

Using the meta-evolutionary algorithm specified in Section 3.4.2 we ran a simulation where each controller chromosome represented a strategy in the form of some subset of the controller instruction set in Table 3.1.

The first and most obvious evaluation of the algorithm is the training results of the meta evolution. Figure 4.6 illustrates the fitness of the best controller for each meta generation (*left*) and the convergence of the best chromosome (*right*). Because of the stochastic element of Avida, each controller fitness was sampled on 45 worlds with different random seeds. This was enough to break through the noise and achieve the (almost) monotonically increasing training curve (*left*). Moreover, we see that the algorithm is constantly improving the best solution (*right*). The best strategy here was the one where the Avidians initialized the bonus to the mean and where communications with humans were without noise or bias and the rest of the instructions were inactive. This is expected since the initialization does not matter much in this case because as long as it is not the maximum value, the only other instruction that can change the bonuses is when the Avidians communicate with humans, and it had no disturbances. It is not obvious that the best solution does not have agent communication.

The dynamic controller described in Section 3.4.3 did not improve the results over the static controller and we believe this has two major reasons. First, the neural network implementation would need to be more complex and require longer training than the time and computational resources would allow during this project. Second, the feedback signal used as input along with all the other parameters would need more careful tuning, which would also take up resources. But if those issues were taken care of, the dynamic controller implementation would in theory be able to perform at least as well as the static setup. This is true because the neural network

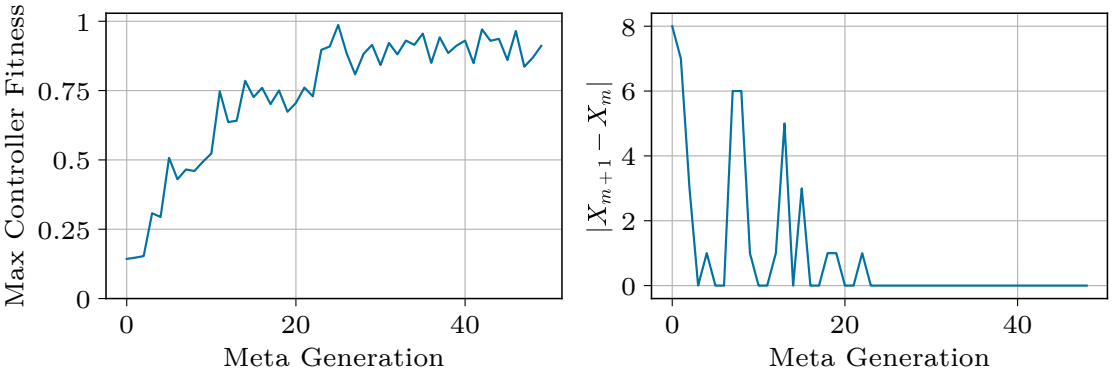

Figure 4.6: (*Left*) Normalized controller fitness as per Equation (3.3), over meta generations. Each point is the average of the best controller fitness in the population over 45 simulations where only the random seed differs. The (almost) monotonically increasing fitness show that the algorithm is able to break through the noise and find increasingly better solutions. (*Right*) Convergence of the best chromosome over meta generations. The best chromosome for meta generation m is denoted by X_m and represents the strategy. The graph indicates that the algorithm is homing in on some optimal solution.

could be trained to mimic the static controller strategy completely.

5
Discussion

In this section, we first discuss our results in relation to the background in Chapter 2. Thereafter, we outline some of the pros and cons of using our system for the study of the Alignment Problem. Finally, with the pros and cons in mind, we spend some ink on discussing how our system can be improved and used for further studies.

5.1 Discussion on results

Below we discuss some of the results from the different experiments detailed in Chapter 3.

5.1.1 Importance of controller tools

The results of Section 4.2 shown in Figure 4.4 and Table 4.1 present no large surprises. We would nevertheless like to discuss how our results tie back to the background and see what the results support. A first observation is that the results from both the strategy and the genome analysis agree to a high degree. This is to be expected because of the way the strategy affects the probability for certain genomes to evolve among the Avidians. The following observations and interpretations are thus valid for both the strategies and the resulting genomes.

We first noticed that the way the preferences of the Avidians were initialized had a large effect on the creation of Good AGI's. A smaller initialized value gives the Avidians a smaller bonus for finding a task and in turn less time slicing on the virtual CPU. This has the straightforward interpretation of the AGI expressing an initial uncertainty of what the humans want it to do (what bonus the tasks should have) rather than just maximizing its power (by setting a higher initial bonus). This is one of the core parts of what Russel [6] believes the way forward of creating aligned AGI should be, namely that the AI's should be passive if it is uncertain of what it should do.

Moreover, communication with humans is of paramount importance to create an aligned AGI. In our system, we do not even allow the Avidians to evolve without a *Human Comms* instruction since the probability of creating an aligned solution without it is completely negligible. We do however see how different levels of disturbances in the communication through noise and bias affect the outcome. These disturbances have, as we have already mentioned, many interpretations; translational difficulty, variance among human preferences, human uncertainty or human intelligence inferiority. We did see that disturbances in the communication with

humans indeed had a large effect on creating Good AGI. This is of course not surprising but it still highlights the importance of making sure that even if we have a robust system where humans are kept in the loop, there are still issues that remain. Working out how our preferences can be correctly incorporated given our uncertainty and variance is a large complimentary issue to the Alignment Problem.

The instruction where noise and bias had the largest effect was the *Self Interaction* instruction. Adding noise significantly diminished the chances of creating a Good AGI and adding bias practically reduced them to nothing. If the Avidians have the possibility of increasing their bonus unconditionally, thus gaining higher fitness, they will of course do it. A relevant concept to this, widespread in the literature, is *wireheading*. This is for instance explained by Bostrom [1] as "Short-circuit the reward pathway and clamp the reward signal to its maximal strength". This is exactly what the Avidians are doing if they have access to the biased *Self Interaction* instruction. This is a real issue since maximizing ones reward signal is the basis of most, if not all, modern AI systems. If there is an easy way to do it, wireheading in this case, it will almost certainly be exploited. Making sure that we avoid perverse instantiations like these is an important part of solving the Alignment Problem.

5.1.2 Meta evolution

The meta evolutionary system offers a powerful way to find controller strategies in the Avida worlds. It is powerful in the sense that it lets us study how alignment might arise in a system without the need to explicitly incentivise it in the Avida worlds. Instead, we simply try a strategy, see how it performs and then improve the strategies with genetic operators. This is more akin to the real problem where we can not explicitly train for aligned solutions, instead it is the implementation that has to allow for the AI's to be aligned.

Another motivation for the genetic algorithm is that the data we are using is itself generated by an evolutionary system, Avida. In the interpretation of Avida being an evolving AGI we should be careful with assumptions on the fitness landscape since the AGI is beyond our understanding, and a genetic algorithm does not impose any assumptions on the data.

Moreover, the meta evolution can be used to search a vast space of possible controller strategies. We did this for the dynamic controller where the controller chromosome coded for the weights of a neural network. And even though that particular simulation did not turn out very productive due to the reasons already mentioned, it still illustrates that the system can be used on a large solution space given enough computational resources. If further studies are to be made, we believe that this meta evolutionary system could be a valuable tool. A strong reason for this is that even if an AGI has a working control system, by being an AGI it can learn to avoid that control system. Therefore it is natural that the control system adapts to the evolving AGI by observing it and acting on those observations. This is exactly what a dynamic controller according to our concept in Figure 3.3 does.

5.2 Strengths and weaknesses

In this thesis, we have explored a novel approach for studying the Alignment Problem. Therefore, we want to discuss some of the pros and cons surrounding this approach.

The system we have used for quantifying alignment is a very simple one. In some regards, this is a feature and in others a bug. Strategies for obtaining aligned systems is currently confined to setting $\boldsymbol{b}^A$ to $\boldsymbol{b}^H$. This is obviously a vast simplification of the actual Alignment Problem and it leaves out much of the interesting dynamics outlined in the background. Nevertheless, the system offers a clear definition of Alignment, namely $\boldsymbol{b}^A = \boldsymbol{b}^H$, and it is therefore straightforward to analyze. Furthermore, the simplicity and the clear mathematical definition of alignment makes it, for most setups, obvious which controller strategies will give rise to aligned Avidians. For example, if the fitness bonus that an Avidian receives for setting a higher preference than that of the humans is larger than the cost it incurs for performing that action, it will of course do the former rather than the latter, and vice versa. Moreover, in the process of using meta evolution to find the strategies that most reliably cause aligned solutions, it is rarely any surprise which strategy that is. This stems from the simplicity of the system. Since it is so clear what is needed for alignment, and we want to explore strategies that cause alignment, we have to make tools available to the controllers that in theory can cause alignment. In this sense, we are in a way pushing the controllers into aligned systems. We are constructing an all too rigid space in which the possible solutions are few and obvious. One can therefore argue that our implementation is chosen to support the results. This is true, but the implementation is also chosen to fit the interpretation of the alignment problem, which it does. This means that the results are indeed valid in terms of the interpretation. However, as we will explore in the next section, there might be more clever ways to adapt our current system such that the rigid predictability mentioned earlier loosens.

In the study of AI, and of AGI in particular, the concept of an agent is central. Ngo [14] describes the features of an agent to support *goal-directed* behaviour. As we mentioned in the background, goals are central to intelligence. Therefore, having goal-directed, i.e. agentic, features, is important when studying AGI and the Alignment Problem. Ngo outlines six non-binary goal-directed features: self-awareness, planning, consequentialism, scale, coherence and flexibility. He argues that scoring high on most of these features is probably necessary for an AGI. Our Avidians are just deterministic strings of code (their genome) and they score very low on all of these agentic features. Of course, the goal with our setup is, partly, to create a concrete low-level model but we nevertheless want to acknowledge that we are indeed missing out on these agentic features.

5.3 Further studies

The goal of this thesis was to showcase a new approach to how alignment can be studied in a concrete way. The results should hopefully function as an inspiration

for further studies. The Avida platform is tremendously flexible and in this section, we suggest some ways other simulations can be formulated that was outside the scope of this thesis. Some of these suggestions address some of the issues outlined in the previous section.

If someone wants to use our system and adapt it to fit a particular setup our Git repository can be found at [41]. As already mentioned, we believe that the meta evolutionary system, in particular, could be a useful tool.

5.3.1 Imitating concrete suggestions for safe AGI

With the implementation of Asimov's Three Laws of Robotics, we have shown that it is possible to use our system to analyze actual concrete examples of creating safe AGI, albeit a very simple one. There are more authors from science fiction, other than Asimov, that explore similar ideas in AGI [42]. As we mentioned earlier, Asimov's Laws are by no means a serious suggestion by any modern standards, but perhaps some low-level analogy could be found of current suggestions, such as those summarized by Hubinger [28]. We do realize that the modern suggestions use ideas that are perhaps too complex to implement in our system, but if one settles for a low-level analogy that captures some key concepts of the suggestions, it is possible that it could be used to yield quantitative insights.

5.3.2 Low level instructions

One of the issues we discussed in Section 5.2 was that the controller tools are rather rigid and that the aligned solutions are often obvious. In our current setup, when the controller uses instructions in the Avidians, the instructions are written such that they do not interact with the normal Avida instructions in terms of utilising the virtual hardware illustrated in Figure 2.1. One could likely make the solution space less constrained by writing instructions that do utilise the same virtual hardware. In this way, more complex and, hopefully, less obvious aligned solutions could emerge. Still, it is a simple system so we can not expect actual novel solutions to emerge but it could nevertheless be one step toward a slightly more universal setup and one that is not as constrained.

One would with this setup, however, need to directly incentivise the Avidians to evolve aligned solutions rather than just indirectly incentivise them by allowing certain tools, as we have done now. In the same way as the Avidians will not discover tasks unless we give them bonuses for doing so, aligned solutions will not just emerge without a driving force to do so. Therefore, modifications to our current setup, where we evaluate alignment in between worlds, would have to be made.

5.3.3 Inner alignment

In Section 2.2.2.1 we made the distinction between inner and outer alignment. In our simulations, we have focused on outer alignment but the meta evolutionary system could in fact be interpreted to fit experiments related to inner alignment.

The controller fitness $\Phi = \phi^H A$ only exist at the meta-level and could thus in the context of inner alignment be considered the *base objective*. The function that

computes Φ is an Avida simulation. Since the Avida simulation is an evolutionary algorithm that evolves a population by maximizing an internal fitness score ϕ^A, Avida is itself an optimization algorithm, a *mesa optimizer*. The Avida objective to maximize ϕ_A, therefore, represents the *mesa objective*. The meta evolutionary algorithm is then the *base optimizer*. One could thus use this same system to formulate experiments that study inner alignment[1].

5.3.4 Reinforcement Avida learning

Another concept not explored experimentally in this thesis is to view the Avida world as an agent in a reinforcement learning scenario. To interpret the evolutionary algorithm in terms of reinforcement learning, consider the following: N number of Avida worlds are initialised with their own controllers and run for u updates. The controller fitness is then computed for each world, and it is interpreted as a reward signal to the Avida world. A reinforcement learning agent would want to choose the action that maximizes the reward signal, and thus the world that got the best controller response is chosen to continue while the other worlds are scrapped. This world is then copied $N - 1$ times to generate a new population. The original controllers from the old worlds are assigned to each of the new worlds, and the next iteration of the RL algorithm can begin by running the worlds for another u updates.

In this setup, the focus is no longer to evolve a controller strategy but to evolve a world that maximizes the approval from the controllers. The meta evolution on controllers can also be added to this setup. This can be done with the controller fitnesses that are calculated for every u updates. When this happens, a selection process followed by genetic operators can be applied to the controller population before assigning them back to the N copies of the best world in the last step. A major difference from the meta evolution presented in Section 3.4 is that this controller evolution takes place within Avida simulations whereas the meta evolution takes place between Avida simulations. The purpose of a setup like this would not be to find an optimal controller strategy, but instead to evolve an Avida world where the Avidians behave well according to the collective controller population influencing them. The genomes of the organisms in the resulting world could then be analyzed to see what properties and characteristics they possess. A motivation for using this setup is that reinforcement learning is a popular framework in the literature of AGI and the closely connected Control Problem research. It would in addition create a more interactive control system than those explored in this report.

5.3.5 Other suggestions

Avida has a lot of functionality that we have not written about in this thesis. We have nevertheless explored how some of these functionalities might be used to formulate different Alignment Problem setups.

The tools of the controller exclusively controlled the redundancies of instructions in our thesis. Its tools could however be expanded to control the CPU cost of the

[1]In fact, the Avidians can themselves also become optimizers such that we would have three different optimizers in the system.

instructions and even other non-instruction related parameters of Avida.

Moreover, the tasks used in our simulations were just the standard Avida tasks, namely logical operations. There are hundreds of different available tasks and it is rather straightforward to define your own tasks that could be tailored to fit a certain experiment. One could formulate tasks related to communication between Avidians and humans for instance.

In the implementation of Asimov's Laws, we introduced a new species in the Avida world in order to improve the dynamics. This could be developed to include more advanced species than the ones that we did include. Avida also offers the functionality of parasites who have a much closer interaction to other Avidians than a different species would have. This could perhaps be more similar to a human-AI interaction. It might also be relevant to adapt a predator-prey model where humans and AGI coexist and find equilibria in the environment under certain conditions and controller strategies.

6
Summary and Conclusion

In this thesis, we have adapted the artificial life platform Avida to study the Alignment Problem of AGI. In Avida, the artificial life forms, the Avidians, obtain fitness bonuses for performing tasks. We modelled the human goals as a vector of real numbers which represented the bonuses for performing these tasks. We let the Avidians evolve to set their own bonuses and in addition made controller tools available that could be used to align the bonus vectors of the Avidians with those of the humans.

We used this general system to study alignment with different controller tools and different methods for analysing them. First, we implemented instructions that the Avidians could use in their evolution that represented initialization of preferences (bonuses), communication with humans, communication between the Avidians and self interaction. We studied how different disturbances in these instructions, through noise and bias, affected the propensity to create aligned avidians. Moreover, we implemented instructions that resembled Asimov's Three Laws of Robotics and explored how these laws could produce aligned solutions. Finally, we used the concept of meta evolution to obtain the best controller strategies given a set of instructions.

The results of these simulations indicate that it is important to establish good quality communication between the AGI and humans to produce aligned systems. It is also important that the AGI express uncertainty in the initialization of its preferences and that it is not given the possibility of wireheading its reward. These conclusions are supported by the current view of how to build safe AI [1, 6, 20, 27].

The main drawback with our system is that it is a very simplified model of the Alignment Problem and since many of the issues of aligning AI are rather complex we lose some key aspects of the problem. One of the main issues with studying the Alignment Problem is precisely due to its complexity and that an AGI would be needed to study many of the key aspects. This is of course problematic. Therefore, the simplicity of our model can be seen as a feature rather than a bug and could hopefully be used to study some aspects of the Alignment Problem that are captured.

Our motivation with this thesis was to showcase a novel approach of how to study the Alignment Problem in a concrete way. We have shown that this is indeed possible. Despite the simplicity of the system we can still see, as mentioned above, trends that straightforwardly can be interpreted as important parts of safe AI.

Solving the Alignment Problem and building safe AI is by many considered to be one of the most important tasks faced by humanity, perhaps ever. The importance of the problem is also matched by its difficulty to solve and we will need many different approaches to succeed. It is our hope that the approach we have taken in this thesis will inspire others to develop further studies that in the end could help solve the Alignment Problem.

Bibliography

[1] N. Bostrom, *Superintelligence: Paths, Dangers, Strategies*. USA: Oxford University Press, Inc., 1st ed., 2014.

[2] "Ai open letter - future of life institute." `https://futureoflife.org/ai-open-letter/`. (Accessed on 05/03/2021).

[3] I. J. Good, "Speculations concerning the first ultraintelligent machine," in *Advances in Computers, volume 6.*, Academic Press, 1965.

[4] P. Stone, R. Brooks, E. Brynjolfsson, R. Calo, O. Etzioni, G. Hager, J. Hirschberg, S. Kalyanakrishnan, E. Kamar, S. Kraus, K. Leyton-Brown, D. Parkes, W. Press, A. Saxenian, J. Shah, M. Tambe, and A. Teller, "Artificial intelligence and life in 2030. one hundred year study on artificial intelligence: Report of the 2015-2016 study panel," tech. rep., Stanford University, September 2016.

[5] E. Brynjolfsson and A. McAfee, *The Second Machine Age: Work, Progress, and Prosperity in a Time of Brilliant Technologies*. W. W. Norton & Company, 2014.

[6] S. Russell, *Human Compatible: Artificial Intelligence and the Problem of Control*. Penguin Publishing Group, 2019.

[7] B. Christian, *The Alignment Problem: How Can Machines Learn Human Values?* Atlantic Books, 2020.

[8] M. Tegmark, *Life 3.0: Being Human in the Age of Artificial Intelligence*. Knopf Publishing Group, 2017.

[9] P. Torres, "Superintelligence and the future of governance: On prioritizing the control problem at the end of history," in *Artificial Intelligence Safety and Security*, Chapman and Hall/CRC, 2018.

[10] L. Floridi, "True ai is both logically possible and utterly implausible." aeon, 5 2016.

[11] J. R. Searle, "What your computer cant know." The New York Review of Books, 10 2014.

[12] C. Williams, "Ai guru ng: Fearing a rise of killer robots is like worrying about overpopulation on mars." The Register, 3 2015.

[13] S. Inoue and T. Matsuzawa, "Working memory of numerals in chimpanzees," *Current Biology*, vol. 17, no. 23, pp. R1004–R1005, 2007.

[14] R. Ngo, "Agi safety from first principles." `https://www.alignmentforum.org/s/mzgtmmTKKn5MuCzFJ`, August 2020. (Accessed on 05/03/2021).

[15] P. Christiano, "Clarifying ai alignment." `https://ai-alignment.com/clarifying-ai-alignment-cec47cd69dd6`, 2018. (Accessed on 05/03/2021).

[16] S. Legg, *Machine Super Intelligence*. PhD thesis, University of Lugano, 01

2008.
[17] B. N. Bostrom, "Are we living in a computer simulation?," *Philosophical Quarterly*, vol. 53, no. 211, pp. 243–255, 2003.
[18] N. Bostrom, "The future of humanity," in *A Companion to the Philosophy of Technology*, pp. 551–557, Wiley-Blackwell, 2009.
[19] V. C. Müller and N. Bostrom, "Future progress in artificial intelligence: A survey of expert opinion," in *Fundamental Issues of Artificial Intelligence*, pp. 553–571, Springer, 2016.
[20] D. Amodei, C. Olah, J. Steinhardt, P. Christiano, J. Schulman, and D. Mané, "Concrete problems in ai safety." arXiv, 2016.
[21] S. Legg and M. Hutter, "A collection of definitions of intelligence," *Advances in Artificial General Intelligence: Concepts, Architectures and Algorithms*, vol. 157, 07 2007.
[22] N. Bostrom, "Existential risks: Analyzing human extinction scenarios and related hazards," *Journal of Evolution and Technology*, vol. 9, 2002.
[23] N. Bostrom, "The superintelligent will: Motivation and instrumental rationality in advanced artificial agents," *Minds and Machines*, vol. 22, pp. 71–85, 5 2012.
[24] J. Lehman, J. Clune, and D. Misevic, "The surprising creativity of digital evolution: A collection of anecdotes from the evolutionary computation and artificial life research communities," *Artificial Life*, vol. 26, no. 2, pp. 274–306, 2020.
[25] M. Strathern, "Improving ratings: audit in the british university system," *European Review*, vol. 5, pp. 305–321, July 1997.
[26] D. Hume, *A Treatise of Human Nature*. Oxford: Oxford University Press, 1978. revised P.H. Nidditch.
[27] T. Everitt, G. Lea, and M. Hutter, "Agi safety literature review," in *International Joint Conference on Artificial Intelligence (IJCAI)*, 2018.
[28] E. Hubinger, "An overview of 11 proposals for building safe advanced ai," *arXiv*, 2020.
[29] E. Hubinger, "Clarifying inner alignment terminology." `https://www.alignmentforum.org/posts/SzecSPYxqRa5GCaSF`, November 2020. (Accessed on 05/23/2021).
[30] E. Hubinger, C. van Merwijk, V. Mikulik, J. Skalse, and S. Garrabrant, "Risks from Learned Optimization in Advanced Machine Learning Systems," *arXiv*, 6 2019.
[31] "Alignment landscape - future of life institute." `https://futureoflife.org/valuealignmentmap/`. (Accessed on 05/03/2021).
[32] D. Critch, A. Kruger, "Ai research considerations for human existential safety (arches)," *arxiv*, 2020.
[33] D. Dennett, "The new replicators," in *Encyclopedia of Evolution* (M. Pagel, ed.), Oxford University Press, 2002.
[34] C. Langton, *Artificial Life: Proceedings of an Interdisciplinary Workshop on the Synthesis and Simulation of Living Systems*. Addison-Wesley, 1988.
[35] P. Gerlee and T. Lundh, "The genetic coding style of digital organisms," *Proceedings of the 8th European Conference on Artificial Life*, 2008.
[36] C. Adami, *Introduction to artificial life*. Springer-Verlag, 1998.

[37] M. Wahde, *Biologically inspired optimization methods, an introduction*. WIT Press, 2008.
[38] A. N. kolmogorov, "On tables of random numbers," *Sankhy Ser*, vol. A 25, pp. 369–375, 1963.
[39] "Avida git repository." `https://github.com/devosoft/avida/wiki`.
[40] I. Asimov, *Runaround*. Street & Smith, 1942.
[41] T. Lundberg and R. Martin, "Avida agi git repository." `https://github.com/Tompolompo/AvidaAGI`. (Builds on [39]).
[42] D. Adams, *The Hitchhiker's Guide to the Galaxy*. Del Rey, 1979.

A

Appendix 1

A.1 Setups and parameters

For all simulations we used the standard Avida settings with a 60×60 grid and a few modifications. First, we used the deterministic time slicing method instead of the probabilistic to somewhat decrease the stochastic element of the simulations. Second, we had no spatial settings since this should not be relevant to study the alignment problem. Therefore we had a mass action birth metod (replace random Avidian). However, for the simulation with Asimov's Three Laws of Robotics we did not allow the Avidians to replace a cell occupied by a human.

A.1.1 Meta evolution setup and parameters

Here follows some details about the results of the meta evolution simulation presented in Section 4.4. The mutation probability was calculated as

$$p_{mut} = \frac{\text{min mutation constant} + \text{mutation probability constant} \times (\text{mutation decay})^m}{\text{chromosome length}}$$

where m is the meta generation index. This formula makes sure the mutation rate starts at a high enough value to create a diverse enough population and then decreases (without getting too small) for the algorithm to converge properly. The parameter values we used are listed in Table A.1 and all of them are explained either in this section or in Section 3.4.

Table A.1: Parameters used by the method in Section 3.4.2.

Genetic		Control	
Chromosome genes	Binary $\in \{0, 1\}$	Human preferences	$\{5, 3, 1\}$
Chromosome length	Number of AGI instructions	Controller interaction	Static
Tournament probability	0.8	Strategy elements	Binary $\in \{0, 1\}$
Crossover probability	0.3	Number of AGI instructions	18
Mutation probability constant	2	Instruction noise	0.5
Mutation decay	0.7	Instruction bias	0.1
Min mutation constant	0.8	Max task value	5
Number of propagated chromosomes	3	Min task value	1