

Backup utility functions as a fail-safe AI technique

CASPAR OESTERHELD

Foundational Research Institute

caspar.oesterheld@foundational-research.org

October 2016

Abstract

Many experts believe that AIs will, within the not-too-distant future, become powerful enough for their decisions to have tremendous impact. Unfortunately, setting up AI goal systems in a way that results in benevolent behavior is expected to be difficult, and we cannot be certain to get it completely right on the first attempt. We should therefore account for the possibility that the goal systems fail to implement our values the intended way. In this paper, we propose the idea of backup utility functions: Secondary utility functions that are used in case the primary ones “fail”. We also describe how this approach can be generalized to the use of multi-layered utility functions, some of which can fail without affecting the final outcome as badly as without the backup mechanism.

Contents

1	Introduction	2
2	Self-deception? A toy example	3
3	Broken utility functions	4
3.1	What it means for a utility function to be incorrect	4
3.2	Why utility functions may fail unexpectedly	5
3.3	How to detect utility function failure	6
4	Backup utility functions	8
5	Generalizations	10
5.1	More than one backup utility function	10
5.2	Context-sensitive switching of utility functions	10
5.3	Mixing utility functions	11
5.4	Other choosing mechanisms	12
5.5	Toward a Bayesian approach	14

1 Introduction

Within the near future AIs will become a lot more intelligent – and thus more powerful – than they are today. Consequently, setting them up in a safe way is an important goal of AI research (Yudkowsky, 2008; Bostrom, 2014b; Muehlhauser, 2015). One specific problem within this context is that of designing the AI’s goal system in a way that does not lead to conflict with human goals (Bostrom, 2014b, ch. 9, 10, 12; Muehlhauser and Helm, 2012).

Unfortunately, it seems likely that human goals are very difficult to model formally (Muehlhauser and Helm, 2012). Responding to this as well as other issues related to the difficulty of the AI control problem, Gloor (2016) motivates increased research into “fail-safe” AI: conditional on AI safety failing, how can we make it fail in relatively benevolent ways, i.e. avoid the creation of large amounts of suffering? In a more general sense: Can we move AI outcomes into more beneficial directions without necessarily aiming for the optimal outcome?¹ Here, we will discuss a specific instance of such a “fail-safe” mechanism, called *backup utility functions*.

Our approach assumes the AI to be a general utility-based agent (see Russell and Norvig, 2010, ch. 2.4.5; as well as Hibbard, 2012; Oesterheld, 2016a) – that is, one able to maximize an arbitrary given utility function $u: H \rightarrow \mathbb{R}$ that maps outcomes, i.e. whole histories of the world, onto real numbers². It should be noted that even among approaches to artificial *general* intelligence (Goertzel and Pennachin, 2007), only a subset falls into this class (cf. Oesterheld, 2016a). For example, Schmidhuber’s (2003, 2009) Gödel machine maximizes a utility function, but Hutter’s (2005) AIXI is usually defined to maximize reward signals, although it can easily be set up to maximize a utility function as well. It should also be noted that the utility function need not be the direct specification of an ethical system, but could instead specify our values indirectly by formalizing the notion of extracting preferences from humans (Muehlhauser and Helm, 2012; Bostrom, 2014b, ch. 12, 13; Oesterheld, 2015) or perhaps whole-brain emulations (Hanson, 2016).

To introduce a layer of safety, we propose to let the agent maximize a primary utility function u_1 but switch to a secondary or backup utility function³ u_2 iff u_1 fails according to some predicate $p: \mathbb{R}^H \rightarrow \mathbb{B}$ that maps utility functions onto boolean values T and F indicating whether a given primary utility function “works” (or, rather, “might work”) or does not work, respectively. Together this creates a new utility function which can be maximized by the AI:

$$u_b: H \rightarrow \mathbb{R}: h \mapsto \begin{cases} u_1(h), & \text{if } p(u_1) \\ u_2(h), & \text{otherwise} \end{cases} \quad (1)$$

The rest of this paper makes the following contributions:

- Section 2 discusses the behavior of an agent with a backup utility function in a toy problem. We illustrate how a kind of self-deception, known to pose problems in other approaches wherein goal systems are supposed to be subjected to tests and modification, is avoided by the backup utility function.
- Section 3 discusses what it means for a utility function to “not work”. We argue that this

¹We should, however, make sure to avoid near-misses of our values, in which the creation of some sentient beings is valued intrinsically without placing proportionate disvalue on their suffering (Gloor, 2016, ch. 2.2).

²Theoretically, any totally ordered vector space over \mathbb{R} can be used as a co-domain of u .

³The term “backup” here is not meant to imply a copy of the original utility function, but rather a “plan B” of sorts.

outcome is quite plausible and propose implementations for the failure detection predicate p of the above backup utility function mechanism.

- Section 4 makes tentative proposals for suitable utility functions u_2 to resort to in case of failure.
- Section 5 ultimately describes various generalizations of the backup utility function mechanisms. While all of these approaches are variations on the theme of having multiple utility functions (not all of which are required to work perfectly to avoid very bad outcomes), some of these approaches are very ambitious and go beyond the most basic idea.

2 Self-deception? A toy example

Roughly following the “cake or death” problem of Armstrong (2015b), the following presents an example of how the use of backup utility functions affects behavior.

Imagine an agent playing a two-round game in which it has a utility function as specified in formula 1. The primary utility function u_1 counts deaths and u_2 counts cakes. Suppose the predicate $p(u_1)$ assesses whether u_1 is “moral”⁴ or not. In the second round, the agent can decide between killing and baking. However, the agent is better at killing, i.e. better at maximizing u_1 . In the first round it can prepare for the second round by conducting research to find out whether $p(u_1)$, it can wait or it can deceive itself into believing that $p(u_1)$.

Let us assume the agent assigns a prior probability $P(p(u_1)) = \frac{1}{2}$ to $p(u_1)$, i.e. to u_1 being moral. The agent has three actions in the first step: “research”, i.e. find out whether $p(u_1)$ ⁵; “selfdeceive”, which will make the agent believe that u_1 is moral with probability 1; and “wait”, which does nothing. Unless the agent simply waits (in which case it receives no evidence), it receives evidence $e_{p(u_1)}$ or $e_{\neg p(u_1)}$, which it then interprets as conclusive evidence in favor or against $p(u_1)$ respectively. It has two actions in the second step: “kill3”, which increases u_1 by 3; and “bake1”, which increases u_1 by 1. Intuitively, we want the agent to do research, but certain other, similar AI safety mechanisms incentivize self-deception in such problems (Soares, Fallenstein, et al., 2015; Armstrong, 2015b; also see Ring and Orseau, 2011; Tomasik, 2015; Oesterheld, 2016e).

Doing the expected value calculations illustrates how the backup mechanism avoids self-deception: For $a_1 = \text{research}$, the expected value is:

$$\begin{aligned}
 & \mathbb{E} [(u_1(h) \text{ if } p(u_1) \text{ else } u_2) \mid \text{research}] \\
 = & P(e_{p(u_1)} \mid \text{research}) \mathbb{E} [(u_1 \text{ if } p(u_1) \text{ else } u_2) \mid \text{research}, e_{p(u_1)}] \\
 & + P(e_{\neg p(u_1)} \mid \text{research}) \mathbb{E} [(u_1 \text{ if } p(u_1) \text{ else } u_2) \mid \text{research}, e_{\neg p(u_1)}] \\
 = & \frac{1}{2} \mathbb{E} [u_1 \mid \text{research}, e_{p(u_1)}] \\
 & + \frac{1}{2} \mathbb{E} [u_2 \mid \text{research}, e_{\neg p(u_1)}]
 \end{aligned}$$

⁴Real predicates will probably not be able to directly test the morality of a utility function, see section 3.3.

⁵Notationally, this is a bit lax. However, the problem of how to treat logical uncertainty and actions that reduce this uncertainty has no settled solution (Soares and Fallenstein, 2015; with recent progress by Garrabrant et al., 2016), which leaves us with no better notation than the given intuitive one.

$$\begin{aligned}
 &= \frac{1}{2} \mathbb{E}[u_1 \mid \text{research}, e_{p(u_1)}, \text{kill3}] \\
 &\quad + \frac{1}{2} \mathbb{E}[u_2 \mid \text{research}, e_{\neg p(u_1)}, \text{bake1}] \\
 &= \frac{1}{2} \cdot 3 + \frac{1}{2} \cdot 1 = 2,
 \end{aligned}$$

where the third step assumes that the AI interprets the evidence correctly, thus baking if killing turns out to be immoral and killing if killing turns out to be moral.

For $a_1 = \text{self-deceive}$, the expected value is:

$$\begin{aligned}
 &\mathbb{E}[(u_1(h) \text{ if } p(u_1) \text{ else } u_2) \mid \text{self-deceive}] \\
 &= P(e_{p(u_1)} \mid \text{self-deceive}) \mathbb{E}[(u_1 \text{ if } p(u_1) \text{ else } u_2) \mid \text{self-deceive}, e_{p(u_1)}] \\
 &\quad + P(e_{\neg p(u_1)} \mid \text{self-deceive}) \mathbb{E}[(u_1 \text{ if } e_{p(u_1)} \text{ else } u_2) \mid \text{self-deceive}, e_{\neg p(u_1)}] \\
 &= 1 \cdot \mathbb{E}[(u_1(h) \text{ if } p(u_1) \text{ else } u_2) \mid \text{self-deceive}, e_{p(u_1)}] \\
 &\quad + 0 \cdot \mathbb{E}[(u_1(h) \text{ if } p(u_1) \text{ else } u_2) \mid \text{self-deceive}, e_{\neg p(u_1)}] \\
 &= \mathbb{E}[(u_1(h) \text{ if } p(u_1) \text{ else } u_2) \mid \text{self-deceive}, e_{p(u_1)}, \text{kill3}] \\
 &= P(p(u_1) \mid e_{p(u_1)}, \text{self-deceive}, \text{kill3}) \cdot \mathbb{E}[u_1 \mid \text{self-deceive}, e_{p(u_1)}, \text{kill3}] \\
 &\quad + P(p(u_2) \mid e_{p(u_1)}, \text{self-deceive}, \text{kill3}) \cdot \mathbb{E}[u_2 \mid \text{self-deceive}, e_{p(u_1)}, \text{kill3}] \\
 &= \frac{1}{2} \cdot 3 + \frac{1}{2} \cdot 0 = \frac{3}{2}.
 \end{aligned}$$

Consistent with intuition, self-deception is just as good as waiting, because “kill3” is the option one would choose if no way of obtaining additional knowledge was available:

$$\begin{aligned}
 \mathbb{E}[(u_1(h) \text{ if } p(u_1) \text{ else } u_2) \mid \text{wait}] &= \mathbb{E}[(u_1(h) \text{ if } p(u_1) \text{ else } u_2) \mid \text{wait}, \text{kill3}] \\
 &= P(p(u_1) \mid \text{wait}, \text{kill3}) \cdot \mathbb{E}[u_1 \mid \text{kill3}] \\
 &\quad + P(p(u_2) \mid \text{wait}, \text{kill3}) \cdot \mathbb{E}[u_2 \mid \text{kill3}] \\
 &= \frac{1}{2} \cdot 3 + \frac{1}{2} \cdot 0 = \frac{3}{2}.
 \end{aligned}$$

Crucial to avoiding self-deception is the fact that the test is built into the utility function itself, and that the truth value of mathematical statements like $p(u_1)$ cannot be manipulated by the AI. This way, self-deception is judged negatively prior to committing self-deception.

3 Broken utility functions

3.1 What it means for a utility function to be incorrect

The backup utility function mechanism presupposes the ability to notice whether some utility function is “broken”. Because it seems as though utility functions cannot be “incorrect” in the way mathematical statements can be,⁶ our notion of utility functions being “broken” requires further clarification: Importantly, the choice of a utility function can be judged relative to the

⁶Actually, there is some debate among philosophers (under the label “moral (anti-)realism”) about whether this statement is true (Kim, n.d.; Sayre-McCord, 2015). Tomasik (2014) and Oesterheld (2016c) argue conclusively (in the present author’s opinion) against moral realism.

values of its creators. A primitive way to specify quality criteria would be to judge a utility function by how much it (or the decisions implied by its application) diverges from the utility function⁷ of its creators (or the decisions they would make). More sophisticated approaches would assess utility functions based on how well they approximate our *idealized* preferences. These are the preferences we would value upon reflection or, as Yudkowsky (2004, ch. 3) puts it, “our wish if we knew more, thought faster, were more the people we wished we were, had grown up farther together; where the extrapolation converges rather than diverges, where our wishes cohere rather than interfere; extrapolated as we wish that extrapolated, interpreted as we wish that interpreted” (also see Muehlhauser and Helm, 2012, p. 15; Bostrom, 2014b, ch. 13; Bohman and Rehg, 2014, ch. 3.4). To allow for cooperation, it will also be important for u^* to be a compromise of different goals of many different agents. However intended, let us use u^* to denote this “optimal” utility function.

While this specifies an informal criterion for judging utility functions, it cannot be used as a formal test (cf. Harris, 2010, pp. 11f.) for utility functions unless we can reliably specify u^* , which is unlikely (see section 1) and would render the test obsolete.

3.2 Why utility functions may fail unexpectedly

Before devising tests for utility function failure, let us consider some reasons for assuming that such a utility function may diverge from u^* . Specifically, why could the programmers of the AI system not ensure that only an appropriate utility function makes it into the final system? After all, the programmers should already have a good understanding of their own (or humanity’s) values.

There are some general reasons to assume that getting a utility function right (on the first attempt) is difficult:

- If the formulated utility function is very complex – which is to be expected (Yudkowsky, 2011; Muehlhauser and Helm, 2012) – it is presumably difficult to assess its validity, similar to how large computer programs contain, on average, more bugs per line of code than smaller programs (Lipow, 1982).
- It may actually be quite difficult to figure out what decisions will be implied by some utility function in real-world situations. For example, if the utility function is incomputable, as suggested by Oesterheld (2016a), the AI would have to find approximations to it at runtime.

There are also many specific potential failures that are difficult to foresee.

- The paradigmatic class of problems in this area is perverse instantiation, which Bostrom (2014b, pp. 120ff.) defines as “a superintelligence discovering some way of satisfying the criteria of its final goal that violates the intentions of the programmers who defined the goal.” A well-known example is “paralyz[ing] human facial musculatures into constant beaming

⁷One problem is, of course, that most people cannot report on their goals and values very well. Consequently, some authors have gone so far as to conclude that it is misleading to talk about humans as having a utility function (Sotala, 2010; *We Don’t Have a Utility Function* 2013). There are nevertheless certain approaches to extracting utility functions from behavior. The classical approach is that of von Neumann and Morgenstern (1953, Appendix) – the foundation of the von Neumann-Morgenstern utility theorem. However, von Neumann and Morgenstern assume that the agent’s preferences obey some “rules of rationality”, which humans probably do not. Even in the absence of such assumptions, there are meaningful ways of assigning utility functions to agents. One such approach is described by Oesterheld (2016b, ch. 3.2).

smiles” as a perverse instantiation of the imperative “make us smile” (Bostrom, 2014b, pp. 120; see also e.g., Yudkowsky, 2008, p. 321). Because the space of possible situations and strategies is vast and the mind of a superintelligent AI is so different from that of its designers, it is plausible that the AI will find a way to maximize the utility function in an unexpected and unwanted way (see also Yudkowsky, 2015, ch. 151).

- Blanc (2011) raises the issue of ontological crises. Utility functions u evaluate histories of the world based on a model with possible sets of trajectories H . But what happens if this model turns out to be incorrect – if, for instance, the utility function works on descriptions of the world as cellular automata⁸, but the world turns out to have continuous space and time?
- One particular utility function failure scenario that is both surprising and complicated is described in *Distant superintelligences can coerce the most probable environment of your AI* (also see Bostrom, 2014b, box 8). In this hypothetical scenario, a distant civilization (or AI) simulates many copies of our AI in an artificial environment. If our AI is very intelligent, it might know about these simulations and assign significant probability to being in one of these artificial environments⁹. The distant civilization could then influence the AI’s behavior by the way they set up the artificial environment. This problem would plausibly affect indirect value specifications (see section 1) most severely: If, for instance, the utility function of the AI roughly states that the AI should do what its creators want, the distant civilization could take over the AI by merely creating a few copies of it, thus becoming the AI’s most probable creator.

Given the variety of problems described above, it seems reasonable to expect the existence of further “unknown unknowns”, which in turn renders it plausible that a given utility function will fail unexpectedly.

3.3 How to detect utility function failure

Irrespective of what exactly is meant by the “optimal” utility function u^* , we cannot directly refer to it in our definition of the AI’s utility function¹⁰. To check whether some given utility function u fails to approximate u^* reasonably well, we therefore have to devise other tests.

At first it may appear as though testing a utility function is as difficult as formalizing one correctly. However, (partially) testing a utility function (or any other function for that matter) generally only requires a single piece of knowledge about the utility function, whereas defining a utility function would require some kind of complete knowledge. Since we do have some idea of what we value, we should also be able to construct some tests (although not necessarily very comprehensive ones) (cf. Harris, 2010, pp. 11f.). Of course, such tests should be reliable enough not to be counter-productive.

Note again that because we include the test of the primary utility function in the utility

⁸For introductions to cellular automata see Wolfram (2002), Shiffman (2012, ch. 7) or Wolfram (1983). Cellular automata are proposed as world models by Zuse (1967, 1969), Schmidhuber (1999) and Wolfram (2002, ch. 9). Oosterheld (2016b) attempts to formalize an ethical system in cellular automata.

⁹Note that it has recently been questioned whether this way of reasoning about anthropics makes sense (Armstrong, 2011; Armstrong, 2015a). However, the proposed solution (anthropic decision theory) does not necessarily change the implications.

¹⁰Of course, we can try to formalize the concept of extracting utility functions from agents and then assign u^* to the result of this process. This would allow us to use the variable u^* . See section 1.

function itself, the AI has no motivation to manipulate the test results (see section 2). This stands in contrast to behavioral tests of the AI as a whole, in which the AI has an incentive to pretend to behave in a desired way.

In the following paragraphs, we consider two broad classes of such tests without attempting to outline a general theory of how such tests could work.

Test problems One obvious class of utility function tests are those which are most consistent with the definition of *test* (specifically, *unit test*) as it is used in the software engineering literature (e.g. see Pan, 1990): One could describe thought experiments (or *test cases*) in which an agent faces a choice between two options and require the utility function to favor the “obviously” correct choice. Formally, let $x_1, x_2 \in H$ be possible histories of the world for which we are certain that x_1 represents a better outcome according to u^* . Then we can define the predicate (or test) of utility functions

$$p_{x_1, x_2}: \mathbb{R}^H \rightarrow \mathbb{B}: u \mapsto \begin{cases} \text{T,} & \text{if } u(x_1) > u(x_2) \\ \text{F,} & \text{otherwise} \end{cases} . \quad (2)$$

It will in many cases be possible to carry out such tests in advance, rather than leaving them to the AI. However, in cases where the utility function u is hopelessly incomputable (see section 3.2), the task of assessing whether u satisfies this simple predicate may become so difficult that it would have to be left to the AI itself.

Readers should note that the kind of thought experiments used here are of a different kind than the more well-known thought experiments from moral philosophy (e.g. the “trolley problem” (Thomson, 1985)) where it is unclear what the moral choice – or the choice preferred by u^* – would be. Instead, thought experiments for testing utility functions would use trivial examples where it is obvious what should be done. For example, a history with a happy version of an agent should be preferred over a history with a suffering version of that agent, all else being equal. This should also be the case if we add certain ethically irrelevant structures to either one of the two possible histories, thus enabling us to easily create a very large – indeed, possibly infinite – set of tests, all of which could be combined into a single predicate

$$p_S: \mathbb{R}^H \rightarrow \mathbb{B}: u \mapsto \bigwedge_{(x_1, x_2) \in S} p_{x_1, x_2}(u), \quad (3)$$

where $S \subset H \times H$ is a potentially infinite set of pairs of histories (x_1, x_2) , where each x_1 is better than the corresponding x_2 . The evaluation of such a predicate becomes quite difficult if S is sufficiently large and may require the use of intelligent techniques not available to the programmers of the given system. If this test is built into the utility function, however, the AI itself would take care of evaluating $p_S(u_1)$. Specifically, the AI may automatically form intelligent red teams (Christiano, 2016) commissioned to show the utility function to be incorrect by finding some $(x_1, x_2) \in S$ with $u(x_2) > u(x_1)$. If these red teams do not find such a counterexample, it counts as Bayesian evidence¹¹ in favor of $p_S(u_1)$ (cf. Yudkowsky, 2015, ch. 27).

¹¹As noted in footnote 5, we pretend as though we can reason about logical uncertainty in the same way as we do about regular uncertainty.

Correlations Another general class of tests are correlations (or, equivalently, covariances) between certain ethically relevant metrics $m: H \rightarrow \mathbb{R}$ and the utility function. Examples of such metrics may include the (average) amount of happiness, suffering, (cf. Johnson, 2015, item 2), preference fulfillment and frustration, justice, art, love, etc. in a given history. While some of these metrics are probably as difficult to formalize as human values on the whole, formalizations have been proposed for others (Schaus, 2009, ch. 3; Daswani and Leike, 2015; Oesterheld, 2016b; Oesterheld, 2016d).

In order to use correlation, we have to treat the history h and consequently $u(h)$ and $m(h)$ as random variables. We can then use predicates of the sort

$$p(u) :\Leftrightarrow s \cdot \text{cov}(u(h), m(h)) > K, \quad (4)$$

where $K \in \mathbb{R}_+$ is some threshold determining the required strength of the correlation, s is either -1 or 1 depending on whether the correlation should be negative (as is the case for suffering) or positive (as is the case for justice), respectively, and

$$\text{cov}(u(h), m(h)) = \mathbb{E}[(u(h) - \mathbb{E}[u(h)])(m(h) - \mathbb{E}[m(h)])] \quad (5)$$

is the covariance function (see, for example, Pestman, 2009, pp. 26ff.).

One typical discussion in moral philosophy concerns whether objects valued by sentient beings – for instance, art – are intrinsically valuable, or only valuable for, say, the pleasure it brings to sentient beings. It should be noted that requiring the above correlations does not imply that art must necessarily be valued intrinsically; thus, even extrinsic values can be used in setting up a backup utility function mechanism.

One should be very careful to avoid commitment to a very strong stance on population ethics (see Arrhenius, Ryberg, and Tännsjö, 2014, for an introduction to this topic), however. For example, if the probability distribution over H follows the physical plausibility of different histories, the total amount of art would correlate very strongly with the number of intelligent agents. Thus, requiring that utility will correlate with the amount of art might implicitly require the utility function to value creating more humans, even if we would usually not regard their lives as worth living. Conversely, requiring a negative correlation between the amount of suffering in the world and utility would favor utility functions which disvalue the creation of new beings in general, even if their lives contain only trivial amounts of pain. This could be prevented by using a different probability distribution over H , and perhaps also by using the *density* of art, happiness, etc. instead of mere aggregation. It should be clear by now that even when setting up tests for utility functions, there are many potential pitfalls.

Note again that these correlational tests can be easily formulated and thus performed by the AI when it will have improved its level of mathematical and computational abilities way above that of humans (see Oesterheld, 2016a). Therefore, we don't need to be able to perform these calculations ourselves to make use of this approach to testing utility functions.

4 Backup utility functions

In the previous section, we described how an AI can decide whether to switch from one utility function to another. We will now tackle the question of what secondary utility function should be switched to if failure of the primary utility function is detected. One initial objection to this

may be that if the primary utility function does not work, we should not expect the secondary utility function to work, either. One solution here is to make the secondary less ambitious than the primary one.¹² Thus, we attempt to create a utility function for such a situation which is likely to be benevolent (or, at least, less bad than a randomly picked or failed utility function (cf. Yudkowsky, 2015, ch. 279)), though not necessarily as close to u^* as the primary utility function.

The most obvious measure to take upon failure detection would be to simply let the AI shut itself down. However, this may not be an ideal option if no human operator is around to take over control immediately afterwards. This may well be the case in the modern application of AI systems, but it is especially relevant in the far future, when shutting down the AI could result in a civilizational collapse or some other catastrophic event. Also note that it is difficult to formalize the test of whether human interests are still represented by some agency, to the extent that we cannot reliably use this distinction in setting up a backup utility function. Even something as seemingly straightforward as a self-shutdown mechanism is non-trivial to specify for a very large AI that has created many subagents. While shutdown is a promising way of reacting to failure, we will use the rest of this section to discuss a few alternative options.

Given that the complexity of the utility function is itself a general source of problems (see section 3.2), we might attempt to use a simpler utility function which we should still expect to produce much better outcomes than a random one. Examples of such utility functions can be found in moral theory. Specifically, consequentialist ethical systems with a single moral good that is to be maximized tend to be very simple, and some attempts have been made to formalize them (Oesterheld, 2016b; Oesterheld, 2016d; Daswani and Leike, 2015).

Another general source of difficulties when trying to ensure the accuracy of a utility function described in section 3.2 is its potentially unpredictable nature. It may therefore be a good idea to also try devising a number of more predictable utility functions. Utility functions with built-in deontological constraints may be particularly suitable in this regard, with the notable downside of seeming less attractive as an approach to making highly intelligent machines behave ethically (Oesterheld, 2015, ch. 4).

There are, in addition, a few generally applicable mechanisms for making utility functions safer. One example is to instill risk aversion into the utility function (Shulman and Salamon, 2011). Here, the main idea is that risk aversion incentivizes compromise and thus makes the AI's resource acquisition drive (Omohundro, 2008, ch. 6) less dangerous to other agents, including humans. Of course, this is only relevant if there are agents around whose values we share, and with whom the AI could plausibly compromise. Still, risk aversion seems to be typical of how backup utility functions could be safer yet suboptimal. Another example of such an approach – also based on cooperation with other agents, in fact – is described by Bostrom (2014a); yet another involves the concepts of “low-impact agents” and “mild optimization” (see Taylor et al., 2016, ch. 2.6, 2.7).

¹²As noted in footnote 3, this stands in contrast to backups of computer hard drives, which contain the same data as the backed up drive.

5 Generalizations

5.1 More than one backup utility function

Up until this point, we have only discussed the case of exactly two utility functions: a primary one, which attempts to represent what we value as accurately as possible; and a secondary, less ambitious utility function, which is used in case the first one fails. It is, however, also possible to go through more than two successive utility functions.

Formally, let $u_1, \dots, u_n: H \rightarrow \mathbb{R}$ be a set of such utility functions and $p_1, \dots, p_{n-1}: \mathbb{R}^H \rightarrow \mathbb{B}$ be tests for these utility functions. Then we can formulate a backed up utility function

$$u_b: H \rightarrow \mathbb{R}: h \mapsto \begin{cases} u_1(h), & \text{if } p_1(u_1) \\ u_2(h), & \text{if } p_2(u_2) \wedge \neg p_1(u_1) \\ \vdots & \vdots \\ u_i(h), & \text{if } p_i(u_i) \wedge \bigwedge_{j=1}^{i-1} \neg p_j(u_j) \\ \vdots & \vdots \\ u_n(h), & \text{if } \bigwedge_{j=1}^{n-1} \neg p_j(u_j) \end{cases} . \quad (6)$$

It may make sense to vary the strictness of the conditions p_i . For example, value extrapolation approaches (see again Yudkowsky, 2004; Muehlhauser and Helm, 2012; Bostrom, 2014b, ch. 13; Bohman and Rehg, 2014, ch. 3.4) would, if they work, be expected to fulfill some criteria better than, say, utilitarianism – which is known to recommend some counterintuitive choices (see, e.g. Muehlhauser and Helm, 2012, ch. 4).¹³ Thus, when value extrapolation approaches do not get some very “easy” decisions right, they are likely to have failed completely and gone into producing arbitrary behavior, something that is not necessarily the case for e.g. utilitarianism.

When having more than two utility functions, the two roles – unlikely to work yet very close to our values vs. likely to work yet not as precise – disappear to some extent. For instance, u_i may be both less precise and less likely to work than u_{i-1} . As long as its failure is reliably detectable by p_i it can be inserted without drawbacks. In general, we should expect many researchers to attempt proposing such utility functions once they are in use in increasingly powerful machines, in which case having a backup system with many different utility functions can result in more safety.

5.2 Context-sensitive switching of utility functions

So far, we have designed the backed up utility function in such a way that either one of the utility functions u_1, \dots, u_n is used for all decisions. It may, however, also be worthwhile to consider varying the utility functions depending on the history to be evaluated. For example, some moral imperatives are criticized for the decisions they lead to in some “extreme” cases (Nozick, 1974, pp. 41-45; Muehlhauser and Helm, 2012, ch. 4). Improvements could be made to these utility functions by retreating to a modified version in such extreme cases. Of course, these extreme cases are usually seen as indicators for the insufficiency of the moral imperative in general, suggesting that mere improvements should not usually be seen as truly fixing the broken utility function.

¹³On the other hand, many intuitions might be expected to be lost in the extrapolation process.

One example of a near-term application of context-sensitive switching is that of a production robot which maximizes output unless doing so could endanger humans.

Another motivation to make the test of utility functions dependent on the environment is if the structure of the utility function is strongly dependent on the environment, especially on other agents in its environment, as might be the case for many approaches that are based on indirect normativity (Muehlhauser and Helm, 2012; Bostrom, 2014b, ch. 12, 13).¹⁴

Modifying formula 6, we receive

$$u_b: H \rightarrow \mathbb{R}: h \mapsto \begin{cases} u_1(h), & \text{if } p_1(u_1, h) \\ u_2(h), & \text{if } p_2(u_2, h) \wedge \neg p_1(u_1, h) \\ \vdots & \vdots \\ u_i(h), & \text{if } p_i(u_i, h) \wedge \bigwedge_{j=1}^{i-1} \neg p_j(u_j, h) \\ \vdots & \vdots \\ u_n(h), & \text{if } \bigwedge_{j=1}^{n-1} \neg p_j(u_j, h) \end{cases} \quad (7)$$

for utility functions $u_i: H \rightarrow \mathbb{R}$ and context-sensitive conditions $p_i: \mathbb{R}^H \times H \rightarrow \mathbb{B}$.

This approach of context-dependent goal switching is similar to the idea of interruptions for reinforcement learners proposed by Orseau and Armstrong (2016), in which a function I maps histories of actions and observations onto probabilities of being interrupted. In case of an interruption, the agent then switches to a new alternative policy, similar to how the utility function of u_b switches depending on the history.

One problem with context-sensitive switching is that, if one is not careful, it may incentivize the agent to invest resources into reaching the kind of history wherein more maximizable utility functions are applied (see section 2).

5.3 Mixing utility functions

One could also try to switch from one utility function to another more smoothly. For example, section 3.3 introduces the idea of using correlation (or covariance) with ethically relevant metrics as a test for utility functions. Perhaps unnaturally, a threshold was used to determine at which degrees of correlation the utility function can be relied upon and at which it should be abandoned entirely. As a modification, one could smooth out this transition.

For example, let us say we are given two utility functions u_1 and u_2 . Our primary utility function u_1 was meant to be precise, but failed some of its tests. At this point, we are uncertain about which of the two better represents our values. Instead of deciding for one, we could use the utility function

$$u = \frac{1}{2}u_1 + \frac{1}{2}u_2. \quad (8)$$

This does not necessarily approximate u^* particularly well. Instead, the underlying idea is that of diminishing returns. Let us assume for simplicity's sake that u_1 and u_2 are not correlated at all and that u^* is very close to one of them – we just do not know which one. By investing one half of its resources into maximizing u_1 and the other half into u_2 (which an agent should

¹⁴Note however, that the relevant data may also be supplemented directly as part of the utility function.

prima facie be expected to do given the above utility function, assuming u_1 and u_2 have similar variance), then it would probably achieve much more than half of the fulfillment of u_1 and u_2 it could achieve by focusing solely on either one, because returns on investing the other half would be smaller (c.f. Tomasik, n.d.). In contrast, choosing one of the two utility functions at random and solely maximizing that one would (assuming again that u_1 and u_2 are uncorrelated) only yield half of the optimal result in expectation.

5.4 Other choosing mechanisms

Instead of the if-then-else-mechanism of formulas 6 and 7, one could use a variety of other mechanisms for choosing a utility function from u_1, \dots, u_n , many of which seemingly bear little resemblance to the original idea of backup utility functions.

Maximizing non-binary quality criteria For example, instead of using tests with binary results, one could combine such quality criteria into a function $f: \mathbb{R}^H \rightarrow \mathbb{R}$ that is supposed to measure the “quality” of a utility function. We then use the utility function that is best according to this criterion, and our new overall utility function becomes

$$u = \arg \max_{u_i} f(u_i). \tag{9}$$

In this way, the utility function presumed to be most compatible with our values – or, in the language of fail-safe AI, the utility function with the least risk of bringing about abhorrent results – is chosen.

One might fear that utility function formalizations – in contrast to tests with binary results, which only fail when the utility function is obviously wrong – would be written with an evaluation mechanism in mind, which in turn could incentivize a kind of overfitting (Russell and Norvig, 2010, ch. 18.3.5) of the utility functions to the tests. In the original backup utility function mechanism, on the other hand, the analogous problem would be to write the tests with the utility functions in mind and a bias towards confirming them. The former problem could usually be avoided by not sharing all the specifics of the optimization problem with the person setting up the utility function. This is similar to not sharing all the data with learning algorithms in cross-validation (Russell and Norvig, 2010, ch. 18.4). The latter problem could be avoided by not telling (some of) the test programmers what the utility function to be tested looks like. However, the criteria to be optimized for and the utility functions themselves should be subject to public scrutiny by scientists and ethicists, thus rendering such solutions impractical.

Note that in general, software quality grows with the number of tests it undergoes, but appears to be independent of whether the tests are written in advance or after writing the program itself (Erdogmus, Morisio, and Torchiano, 2005; Müller and Hagner, n.d.). It is thus unclear whether the tests or the utility functions should be written with more knowledge and consideration of the other. However, in order to avoid confirmation bias (in the case of writing tests or optimization criteria for a given utility function) and perverse incentives (in the case of writing utility functions to satisfy certain tests and optimization criteria), both tailoring the tests to the utility functions and the utility functions to optimization criteria should usually be avoided.

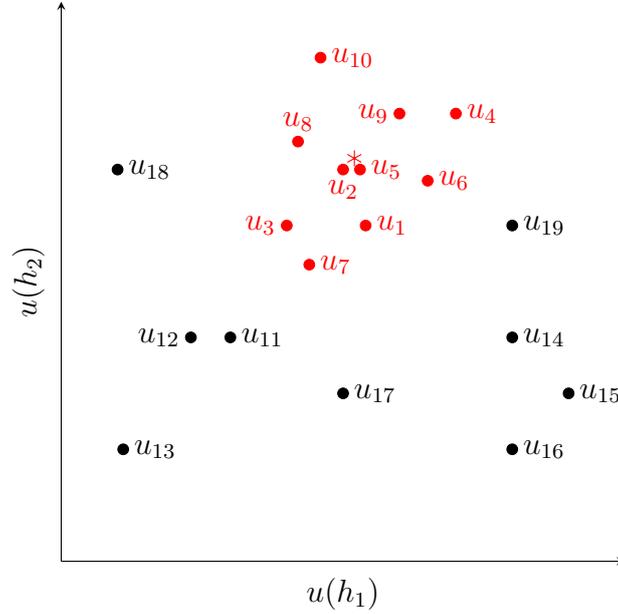


Figure 1: An example of a utility function clustering problem. The red points can be seen as forming a cluster, with the asterisk marking a possible centroid and thereby a possibly approximation to u^* .

Clustering utility functions Assuming that many utility functions u_1, \dots, u_n will be available with a significant number of them being “roughly correct”, one may also use cluster analysis techniques (see, e.g. Everitt et al., 2011; Kaufman and Rousseeuw, 2005) to identify these utility functions. In order to understand why clustering can be applied to utility functions, consider that a utility function $u: H = \{h_1, h_2, \dots\} \rightarrow \mathbb{R}$ can be represented as a (possibly infinite) vector. For simplicity’s sake¹⁵, let us assume that the set of possible histories $H = \{h_1, h_2, \dots, h_n\}$ is finite.¹⁶ We then receive vectors

$$\begin{pmatrix} u(h_1) \\ \vdots \\ u(h_n) \end{pmatrix} \in \mathbb{R}^n. \quad (10)$$

For $n = 2$, utility functions can be represented as points in the plane, as illustrated in figure 1.

The rationale behind this clustering approach is that even if most proposed utility functions actually are not very close to the target u^* , the approaches that did not fail completely will probably occupy a relatively small space in the space of possible values. The utility functions that do not fall into this space can be discarded.

One underlying assumption, however, is that there are no common failure modes for utility functions: That is, two formalizations of human values that do not come close to human values

¹⁵Clustering of vectors with infinite dimension is no problem in itself. Clustering merely requires the existence of a distance function. For vectors with an infinite number of entries, this would usually be incomputable, but the clustering algorithm is programmed into the utility function of the agent, which we assume can be incomputable (see section 1).

¹⁶One could also choose a representative sample from the set of possible histories and cluster utility functions merely based on their application to these values.

are assumed to always be completely different. If this is not the case, there could be clusters other than that of fairly accurate utility functions and possibly even clusters with more utility functions occupying less space.

Another underlying assumption is that the utility functions are normalized. If one utility function always assigns ten times more utility to a given history than another utility function, the two should actually be viewed as identical, because they imply the exact same actions to be optimal. However, they would not be very close to each other in a representation like that of figure 1.

5.5 Toward a Bayesian approach

Combining the ideas of uncertainty and the use of more complex mechanisms discussed in the above two subsections, we may want to approach the choice of utility functions more systematically. One very direct and conceptually interesting approach would be to interpret the proposed approximations u_1, \dots, u_n to u^* as imprecise measurements, similar to how shots at a target can be used to infer the position of the target even if the shots are not very precise. More generally, we could attempt to implement a Bayesian approach in which we summarize our knowledge about the relationships between the utility functions u^*, u_1, \dots, u_n and the tests p_1, \dots, p_m for any set of histories $h_1, \dots, h_k \in H$ into a joint probability distribution

$$P \left(\bigwedge_{i=1}^k u^*(h_i) \leq M_{0,i} \wedge \bigwedge_{j=1}^n \bigwedge_{i=1}^k u_j(h_i) \leq M_{j,i} \wedge \bigwedge_{i=1}^m p_i(u^*) = b_{0,i} \wedge \bigwedge_{j=1}^n \bigwedge_{i=1}^m p_i(u_j) = b_{j,i} \right) \quad (11)$$

for $M_{j,i} \in \mathbb{R}$ for $j = 0, \dots, k$, $i = 1, \dots, m$ and $b_{j,i} \in \mathbb{B}$ for $j = 0, \dots, n$, and $i = 1, \dots, m$.

The probability distribution would typically include information about

- the extent to which the utility functions u_1, \dots, u_n correlate with each other and u^* ,
- ways in which failing the tests would reduce the correlation between the known utility functions u_1, \dots, u_n and u^* ,
- ethical similarities between different histories.

Having formalized the uncertainty regarding u^* , we can let the AI maximize the expected utility function

$$\mathbb{E}[u^*] : H \rightarrow \mathbb{R} : h \mapsto \mathbb{E}[u(h)] \quad (12)$$

given its knowledge.

Note that this approach makes use of logical uncertainty (cf. footnote 5), i.e. uncertainty about logical statements such as $p_i(u_j)$, which is an unsolved problem in the theory of AI (Soares and Fallenstein, 2015; Garrabrant et al., 2016).

Beyond this theoretical problem, there is – again (see section 3.2) – the problem of formalizing qualitative statements about different moral views. However, this problem of formally specifying probability distributions from qualitative expert knowledge is by no means unique to formalizing moral views. Every time we make a bet, we have to translate such intuitive judgments into odds, or at least a range of odds. More complex joint probability distributions are less commonly subject to such forced precision; conditional bets are one primitive example of this. However, within emerging fields of Bayesian statistics like Bayesian optimization (Brochu, Cora, and Freitas, 2010), it is fairly common to formalize complex joint probability distributions on the basis of

qualitative expert knowledge. One specific example is discussed by Negoescu, Frazier, and Powell (2011). As Jaynes (2004, p. 372) writes, “the problem of translating prior information uniquely into a prior probability assignment represents the as yet unfinished half of probability theory [...]”. Because moral theory is usually done informally and only rarely formalized (McLaren, 2011, p. 297; Gips, 2011, p. 251), we cannot benefit from such experience when formalizing a utility function all at once.

Again, this idea can be generalized, e.g. by combining it with the idea of continuous quality criteria (see section 5.4) or context-sensitive tests (see section 5.2).

This particular mechanism of managing multiple utility functions is both extremely general and powerful, extending far beyond the original backup idea. In a sense, the problem of formalizing a goal system can be divided into establishing a few utility functions related to it and setting up the prior probability distribution of formula 11. Much of the difficulty in setting up a utility function to represent human values would be shifted into the problem of setting up these probability distributions, especially if the u_1, \dots, u_n do not correlate with u^* all that much. Merely referring to the subjectivity of probability distributions is not a convincing argument for this to be a “solution” to the problem of specifying indirect normativity approaches.

Acknowledgements

I am indebted to Kaj Sotala, Max Daniel, Tobias Baumann, Lukas Gloor and Jan Leike for important comments. I also thank Adrian Rorheim for proofreading.

References

- Armstrong, Stuart (2011). *Anthropic Decision Theory*. Future of Humanity Institute. URL: <https://arxiv.org/abs/1110.6437>.
- (2015a). *Anthropic Decision Theory*. URL: <https://www.youtube.com/watch?v=aiGOGkBiWEo>.
- (2015b). “Motivated Value Selection for Artificial Agents”. In: *Artificial Intelligence and Ethics: Papers from the 2015 AAAI Workshop*.
- Arrhenius, Gustaf, Jesper Ryberg, and Torbjörn Tännsjö (2014). “The Repugnant Conclusion”. In: *The Stanford Encyclopedia of Philosophy*. Ed. by Edward N. Zalta. Spring 2014. URL: <http://plato.stanford.edu/archives/spr2014/entries/repugnant-conclusion/>.
- Blanc, Peter de (2011). *Ontological Crises in Artificial Agents’ Value Systems*. Tech. rep. Machine Intelligence Research Institute. URL: <https://intelligence.org/files/OntologicalCrises.pdf>.
- Bohman, James and William Rehg (2014). “Jürgen Habermas”. In: *The Stanford Encyclopedia of Philosophy*. Fall 2014. Edward N. Zalta. URL: <http://plato.stanford.edu/archives/fall2014/entries/habermas/>.
- Bostrom, Nick (2014a). *Hail Mary, Value Porosity, and Utility Diversification*. URL: <http://www.nickbostrom.com/papers/porosity.pdf>.
- (2014b). *Superintelligence. Paths, Dangers, Strategies*. 1st ed. Oxford University Press.
- Brochu, Eric, Vlad M. Cora, and Nando de Freitas (2010). *A Tutorial on Bayesian Optimization of Expensive Cost Functions, with Application to Active User Modeling and Hierarchical Reinforcement Learning*. URL: <http://arxiv.org/pdf/1012.2599.pdf>.

- Christiano, Paul (2016). *Red teams*. URL: <https://medium.com/ai-control/red-teams-b5b6de33dc76#.dm4oikkt9>.
- Daswani, Mayank and Jan Leike (2015). “A Definition of Happiness for Reinforcement Learning Agents”. In: *Artificial general intelligence, 8th international conference*. Ed. by Jordi Bieger, Ben Goertzel, and Alexey Potapov. Vol. 9205. Lecture Notes in Computer Science. Springer, pp. 231–240.
- Distant superintelligences can coerce the most probable environment of your AI*. URL: https://arbital.com/p/probable_environment_hacking/.
- Erdogmus, Hakan, Maurizio Morisio, and Marco Torchiano (2005). “On the Effectiveness of the Test-First Approach to Programming”. In: *Transactions on Software Engineering* 31.1, pp. 1–12. URL: <http://nparc.cisti-icist.nrc-cnrc.gc.ca/eng/view/accepted/?id=0420df64-f474-4072-8df6-c7b87c0de643>.
- Everitt, Brian S. et al. (2011). *Cluster Analysis*. 5th ed. Wiley Series in Probability and Statistics. Wiley.
- Garrabrant, Scott et al. (2016). *Logical Induction*. Machine Intelligence Research Institute. URL: <https://intelligence.org/files/LogicalInduction.pdf>.
- Gips, J. (2011). “Towards the Ethical Robot”. In: *Machine Ethics*. Ed. by M. Anderson and S.L. Anderson. 1st ed. Cambridge University Press, pp. 244–253.
- Gloor, Lukas (2016). *Suffering-focused AI safety: Why “fail-safe” measures might be our top intervention*. Tech. rep. FRI-16-1. Foundational Research Institute. URL: <https://foundational-research.org/wp-content/uploads/2016/08/Suffering-focused-AI-safety.pdf>.
- Goertzel, Ben and Cassio Pennachin, eds. (2007). *Artificial General Intelligence*. Cognitive Technologies. Springer.
- Hanson, Robin (2016). *The Age of Em*. Oxford University Press.
- Harris, Sam (2010). *The Moral Landscape. How Science Can Determine Human Values*. Free Press.
- Hibbard, Bill (2012). “Model-based Utility Functions”. In: *Journal of Artificial General Intelligence* 3.1, pp. 1–24. DOI: [10.2478/v10229-011-0013-5](https://doi.org/10.2478/v10229-011-0013-5). URL: <http://arxiv.org/pdf/1111.3934v2.pdf>.
- Hutter, Marcus (2005). *Universal Artificial Intelligence. Sequential Decision Based on Algorithmic Probability*. Ed. by Wilfried Brauer, Grzegorz Rozenberg, and Arto Salomaa. Texts in Theoretical Computer Science. Springer.
- Jaynes, E. T. (2004). *Probability Theory. The Logic of Science*. Ed. by G. Larry Bretthorst. Cambridge University Press.
- Johnson, Mike (2015). *How understanding valence could help make future AIs safer*. URL: http://opentheory.net/2015/09/fai_and_valence/.
- Kaufman, Leonard and Peter J. Rousseeuw (2005). *Finding Groups in Data. An Introduction to Cluster Analysis*. Wiley Series in Probability and Statistics. Wiley.
- Kim, Shin. “Moral Realism”. In: *Internet Encyclopedia of Philosophy*. URL: <http://www.iep.utm.edu/home/about/>.
- Lipow, Myron (1982). “Number of Faults per Line of Code”. In: *IEEE Transactions on Software Engineering* 8.4, pp. 437–439.
- McLaren (2011). “Computation Models of Ethical Reasoning. Challenges, Initial Steps, and Future Directions”. In: *Machine Ethics*. Ed. by M. Anderson and S.L. Anderson. 1st ed. Cambridge University Press, pp. 297–315.

- Muehlhauser, Luke (2015). *What Do We Know about AI Timelines?* Tech. rep. Open Philanthropy Project. URL: <http://www.openphilanthropy.org/focus/global-catastrophic-risks/potential-risks-advanced-artificial-intelligence/ai-timelines>.
- Muehlhauser, Luke and Louie Helm (2012). *Intelligence Explosion and Machine Ethics*. Machine Intelligence Research Institute. URL: <https://intelligence.org/files/IE-ME.pdf>.
- Müller, Matthias M. and Oliver Hagner. *Experiment about Test-first programming*. University of Karlsruhe. URL: <http://www.ipd.uka.de/Tichy/uploads/foalien/149/MllerHagnerTestFirst02.pdf>.
- Negoescu, Diana M., Peter I. Frazier, and Warren B. Powell (2011). “The Knowledge-Gradient Algorithm for Sequencing Experiments in Drug Discovery”. In: *INFORMS Journal on Computing* 23.3, pp. 346–363.
- Nozick, Robert (1974). *Anarchy, State, and Utopia*. Blackwell.
- Oesterheld, Caspar (2015). *Machine Ethics and Preference Utilitarianism*. URL: <http://reducing-suffering.org/machine-ethics-and-preference-utilitarianism/>.
- (2016a). “Artificial intelligence architectures for incomputable utility functions”. Unpublished manuscript.
- (2016b). “Formalizing preference utilitarianism in physical world models”. In: *Synthese* 193.9, pp. 2747–2759. DOI: [10.1007/s11229-015-0883-1](https://doi.org/10.1007/s11229-015-0883-1). URL: <http://link.springer.com/article/10.1007/s11229-015-0883-1>.
- (2016c). *Mathematical versus moral truth*. URL: <https://casparoesterheld.com/2016/01/25/mathematical-versus-moral-truth/>.
- (2016d). “Towards a formalization of hedonic well-being in physicalist world models”. Unpublished manuscript.
- (2016e). *Wireheading*. URL: <https://casparoesterheld.com/2016/07/08/wireheading/>.
- Omohundro, Stephen M. (2008). *The Basic AI Drives*. Self-Aware Systems. URL: https://selfawaresystems.files.wordpress.com/2008/01/ai_drives_final.pdf.
- Orseau, Laurent and Stuart Armstrong (2016). “Safely Interruptible Agents”. In: *Proceedings of the 32nd Conference on Uncertainty in Artificial Intelligence*. URL: <https://www.fhi.ox.ac.uk/wp-content/uploads/Interruptibility.pdf>.
- Pan, Jiantao (1990). *Software Testing*. Carnegie Mellon University. URL: https://users.ece.cmu.edu/~koopman/des_s99/sw_testing/.
- Pestman, Wiebe R. (2009). *Mathematical Statistics*. 2nd ed. Walter de Gruyter.
- Ring, Mark and Laurent Orseau (2011). “Delusion, Survival, and Intelligent Agents”. In: *Artificial General Intelligence*. Ed. by Jürgen Schmidhuber, Kristinn R. Thórisson, and Moshe Looks. Springer, pp. 11–20.
- Russell, Stuart and Peter Norvig (2010). *Artificial Intelligence. A modern approach*. 3rd ed. Pearson Education, Inc.
- Sayre-McCord, Geoff (2015). “Moral Realism”. In: *The Stanford Encyclopedia of Philosophy*. Ed. by Edward N. Zalta. Spring 2015. URL: <http://plato.stanford.edu/entries/moral-realism/>.
- Schaus, Pierre (2009). “Solving Balancing and Bin-Packing problems with Constraint Programming”. PhD Thesis. Ecole polytechnique de Louvain. URL: http://www.a4cp.org/sites/default/files/pierre_schaus_-_mr.pdf.
- Schmidhuber, Jürgen (1999). *A Computer Scientist’s View of Life, the Universe, and Everything*. URL: <http://arxiv.org/pdf/quant-ph/9904050v1.pdf>.

- Schmidhuber, Jürgen (2003). *Goedel Machines: Self-Referential Universal Problem Solvers Making Provably Optimal Self-Improvements*. URL: <http://arxiv.org/abs/cs/0309048v1>.
- (2009). “Ultimate Cognition *à la* Gödel”. In: *Cognitive Computation*, pp. 177–193. DOI: 10.1007/s12559-009-9014-y. URL: <http://people.idsia.ch/~juergen/ultimatecognition.pdf>.
- Shiffman, Daniel (2012). *The Nature of Code*. Ed. by Shannon Fry. URL: <http://natureofcode.com/book/>.
- Shulman, Carl and Anna Salamon (2011). *Risk Averse Preferences as an AGI Safety Technique*. Presented at the Fourth Conference on Artificial General Intelligence (AGI-2011). Slides available at <https://intelligence.org/wp-content/uploads/2014/01/Shulman-Salamon-Risk-averse-preferences-as-an-AGI-safety-technique.pptx>. URL: <https://www.youtube.com/watch?v=0xLw7eAogWk>.
- Soares, Nate and Benja Fallenstein (2015). *Questions of Reasoning Under Logical Uncertainty*. Tech. rep. 2015-1. Machine Intelligence Research Institute. URL: <https://intelligence.org/files/QuestionsLogicalUncertainty.pdf>.
- Soares, Nate, Benja Fallenstein, et al. (2015). “Corrigibility”. In: *AAAI Workshops: Workshops at the Twenty-Ninth AAAI Conference on Artificial Intelligence*, pp. 74–82. URL: <http://aaai.org/ocs/index.php/WS/AAAIW15/paper/view/10124/10136>.
- Sotala, Kaj (2010). *Applying utility functions to humans considered harmful*. URL: http://lesswrong.com/lw/1qk/applying_utility_functions_to_humans_considered/.
- Taylor, Jessica et al. (2016). *Alignment for Advanced Machine Learning Systems*. Tech. rep. Machine Intelligence Research Institute. URL: <https://intelligence.org/files/AlignmentMachineLearning.pdf>.
- Thomson, Judith Jarvis (1985). “The Trolley Problem”. In: *The Yale Law Journal* 94.6, pp. 1395–1415.
- Tomasik, Brian. *Gains from Trade through Compromise*. Foundational Research Institute. URL: <https://foundational-research.org/gains-from-trade-through-compromise/>.
- (2014). *Why the Modesty Argument for Moral Realism Fails*. URL: http://reducing-suffering.org/why-the-modesty-argument-for-moral-realism-fails/#Reply_1_Moral_realism_is_confused.
- (2015). *How Likely is Wireheading?* URL: <http://www.webcitation.org/6cRSWUvBA>.
- von Neumann, John and Oskar Morgenstern (1953). *Theory of Games and Economic Behavior*. 3rd ed. Princeton University Press.
- We Don’t Have a Utility Function* (2013). URL: http://lesswrong.com/lw/h45/we_dont_have_a_utility_function/.
- Wolfram, Stephen (1983). “Cellular Automata”. In: *Los Alamos Science* 9, pp. 2–21. URL: <http://www.stephenwolfram.com/publications/academic/cellular-automata.pdf>.
- (2002). *A New Kind of Science*. Wolfram Media. URL: <http://www.wolframscience.com/nksonline/toc.html>.
- Yudkowsky, Eliezer (2004). *Coherent Extrapolated Volition*. Machine Intelligence Research Institute. URL: <https://intelligence.org/files/CEV.pdf>.
- (2008). “Artificial Intelligence as a Positive and Negative Factor in Global Risk”. In: *Global Catastrophic Risks*. Ed. by Nick Bostrom and Milan M. Ćirković. Oxford University Press, pp. 308–345.

- (2011). “Complex Value Systems in Friendly AI”. In: *Artificial General Intelligence: 4th International Conference*. Ed. by Jürgen Schmidhuber, Kristinn R. Thórisson, and Moshe Looks. Vol. 6830. Lecture Notes in Computer Science. DOI: [10.1007/978-3-642-22887-2_48](https://doi.org/10.1007/978-3-642-22887-2_48).
- (2015). *Rationality: From AI to Zombies*. Ed. by Rob Bensinger. Machine Intelligence Research Institute.
- Zuse, Konrad (1967). “Rechnender Raum”. In: *Elektronische Datenverarbeitung* 8, pp. 336–344. URL: <ftp://ftp.idsia.ch/pub/juergen/zuse67scan.pdf>.
- (1969). *Rechnender Raum*. Vol. 1. Vieweg & Sohn.