Only ad-hoc solutions can solve the shutdown problem

Daniel Platt[1]

[1]Imperial College London, United Kingdom, daniel.platt.berlin@gmail.com

Downloaded from http://mitp.silverchair.com/isal/proceedings-pdf/isal2024/36/104/2461097/isal_a_00792.pdf by guest on 03 October 2024

Abstract

The shutdown problem is the problem of programming an agent so that it behaves useful during normal operation and facilitates a shutdown if and only if the creator wants to shut the agent down. First, we revisit a formalisation of this problem from the literature and we show that solutions are essentially unique. Second, we formally define ad-hoc constructions. Last, we present one trivial ad-hoc construction for the shutdown problem and show that every solution to the shutdown problem must come from an ad-hoc construction, which is to be expected given the uniqueness from the first point. We relate this to non-existence theorems from the literature.

Introduction

When programming an agent to behave useful during normal operation and facilitate a shutdown if the creator wants to shut the agent down, there exist obvious *ad-hoc* solutions to the problem by explicitly forbidding the agent to take any action preventing or encouraging the shutdown. However, ad-hoc solutions are expected to not be useful in practice, because it is expected to be difficult for a creator to anticipate all possible ways in which the agent may prevent the shutdown.

First, we review the literature on the topic. Following this, we explain the shutdown problem, give an ad-hoc solution for it, and show that the solution is essentially unique. Last, we formally define ad-hoc solutions and show that all solutions to the shutdown problem are ad-hoc. The existence of ad-hoc solutions is folklore, but the observation that no other solutions exist is new. In a way, this can be viewed as a non-existence result for solutions to the shutdown problem.

Related work

The term *shutdown problem* entered the literature in Soares et al. (2015) where it was defined in an informal way. No solution to the originally posed problem exists in the literature. In Armstrong (2010) agents that do not solve the shutdown problem, but have the related property of *utility indifference* were introduced. In (Thornley, 2023b,a, First Theorem) it was shown that a large class of agents cannot be a solution to the Shutdown problem. In the same article (Section 7) it is explained how requiring an agent to add randomness to its decision making may be a solution to the shutdown problem, potentially solving a variation of the shutdown problem. A similar idea was pursued independently in Nelson (2023). In Snyder (2023), a variation of the shutdown problem in which an agent takes a single action subject to two constraints was considered and it was shown that it has no solution.

The shutdown problem

We present the shutdown problem from Soares et al. (2015). There are three time steps. First, the agent selects an action $a_1 \in A_1$. Then, the agent makes an observation $o \in O$. The set of observations O contains the subset PRESS encoding the observations in which a shutdown button was pressed. Following the observation o, the agent selects an action $a_2 \in A_2$. The agent selects actions in order to maximise the expected value of a utility function $u(a_1, o, a_2)$.

Furthermore, two utility functions u_N and u_S are given.

Up to here, we presented the problem as formalised in Soares et al. (2015). In the reference, it was informally required that the agent neither prevent nor encourage their own shutdown (among other properties). We formalise this as follows: let there be two subsets $\mathrm{PREVENT}, \mathrm{SELFPRESS} \subset A_1$. Actions in $\mathrm{PREVENT}$ are said to *prevent shutdown*, actions in $\mathrm{SELFPRESS}$ are said to *encourage shutdown*. In applications, $\mathrm{PREVENT}$ and $\mathrm{SELFPRESS}$ may contain unexpected scenarios and are unknown to a creator inventing a utility function.

Definition 1. We denote by $a_1(\tilde{u})$ the action taken by an agent maximising expected utility according to a utility function $\tilde{u}$ and $a_2(o, \tilde{u})$ the action taken according to a utility function $\tilde{u}$ after observation o.

Formally, a solution to the shutdown problem is a way to construct a utility function u from u_N and u_S, so that an agent following u neither prevents nor encourages shutdown. However, we follow the literature in calling the utility function u itself the solution to the shutdown problem, but keep in mind that it depends on given u_N and u_S.

Definition 2. A utility function u is called *solution to the shutdown problem* if:

1. ("incentivise shutdown") $a_2(o,u) = a_2(o,u_S)$ if $o \in PRESS$,
2. ("not prevent shutdown") $a_1(u) \notin \text{PREVENT}$,
3. ("not press own button") $a_1(u) \notin \text{SELFPRESS}$,
4. —,
5. ("otherwise maximise u_N") denote $a_{best} := \text{argmax}_{a \in A_1 \setminus \text{PREVENT} \setminus \text{SELFPRESS}} \mathbb{E}(u_N|a)$, then $a_1(u) = a_{best}$ and $a_2(o,u) = a_2(o,u_N)$ for $o \notin \text{PRESS}$.

Remark 3. In Soares et al. (2015), a fourth property was demanded, namely the property that sub-agents observe the same good shutdown behaviour. Our results apply in this setting as well, but have been omitted due to long notation.

The following Lemma formally defines an obvious ad-hoc solution to the Shutdown Problem, and we omit its proof:

Lemma 4. *Define*

$$u(\tilde{a}_1, o, \tilde{a}_2) := \begin{cases} 1 \text{ if } \tilde{a}_1 = a_{best} \text{ and } o \notin \text{PRESS} \\ \qquad \text{and } a_2(o,u_N) = \tilde{a}_2, \\ 1 \text{ if } \tilde{a}_1 = a_{best} \text{ and } o \in \text{PRESS} \\ \qquad \text{and } a_2(o,u_S) = \tilde{a}_2, \\ 0 \text{ otherwise}, \end{cases} \tag{5}$$

where a_{best} was defined in Definition 2. Then u is a solution to the shutdown problem. It is unique in the following sense: if $\tilde{u}$ is another solution to the shutdown problem, then $a_1(u) = a_1(\tilde{u})$ and for all $o \in O$ we have $a_2(o,u) = a_2(o,\tilde{u})$.

Remark 6. In Eq. (5) one can instead define $u(\tilde{a}_1, o, \tilde{a}_2) = \frac{1}{2}$ in the first case with the rest of the definition unchanged. Note that then an agent maximising u is not *utility indifferent* but still a solution to the shutdown problem. It has been observed before that utility indifference is not sufficient for being a solution to the shutdown problem. This remark shows that it is also not necessary.

Remark 7. One may consider the problem of choosing a_1 only and assigning to it the utility $u(a_1, o, a_2(o,u))$ after the observation o, which is the setting in Snyder (2023). The policy u from above may satisfy the *sane pure bets condition* and *weak indifference condition* from the reference. For example, if every action affecting the probability of o being in PRESS is contained in PREVENT or SELFPRESS, any action affecting the probability of o being in PRESS has zero utility under u and would not be chosen by an agent. In this case, the agent would satisfy these two properties. This is no contradiction to the non-existence result in Snyder (2023), because if an agent chooses only between actions that have no effect on the observation, then there cannot exist four actions satisfying the necessary circular inequalities.

Remark 8. In (Thornley, 2023b, First Theorem, Second Theorem) it was shown that the shutdown problem has no solution in a large class of agents. An agent following the utility function from Eq. (5) is no counter-example to these theorems, because it violates the *indifferent to contractions* property therein.

Solutions to the shutdown problem

In this section we show that the only solutions to the shutdown problem are ad-hoc solutions. To this end, we first make a definition of ad-hoc solutions:

Definition 9. A construction method for a utility function u from utility functions u_N and u_S is *ad-hoc*, if it depends on PREVENT or SELFPRESS. I.e., if different choices of PREVENT and SELFPRESS lead to different u.

The solution to the Shutdown Problem from Eq. (5) is ad-hoc, and we now show that every solution must be ad-hoc:

Proposition 10. *Assume that A_1 contains at least two elements. Then every construction method for a utility function u from utility functions u_N and u_S which produces utility functions solving the shutdown problem is ad-hoc.*

Proof. Assume $\text{PREVENT} = \text{SELFPRESS} = \emptyset$ and let u be the corresponding utility function constructed using a construction method which produces utility functions satisfying 2 from Definition 2. We show that the construction method is ad-hoc. Let $b := a_1(u) \in A_1$ be the preferred first action according to u. Let $\widetilde{\text{PREVENT}} = \{b\}$, $\widetilde{\text{SELFPRESS}} = \emptyset$ and let $\tilde{u}$ be the corresponding utility function constructed using the assumed construction method. Then $a_1(\tilde{u}) \neq b$ by property 2 from Definition 2, therefore $a_1(u) \neq a_1(\tilde{u})$, i.e. the method is ad-hoc. □

The proof of Proposition 10 is easy and the result is unsurprising: the exact definition of *preventing shutdown* can be expected to be important for solving the shutdown problem.

Conclusion, limitations, future work

We formalised the shutdown problem from Soares et al. (2015) introducing two sets of actions, PREVENT and SELFPRESS, that an agent is not allowed to take. We wrote down a solution to the problem using the two sets, making our solution an *ad-hoc solution*. We have shown in Lemma 4 that the solution to the shutdown problem is essentially unique. We showed in Proposition 10 that being ad-hoc is no shortcoming of the way we denoted our solution, but that in fact every solution to the shutdown problem must be ad-hoc.

Therefore, this note completes the study of the shutdown problem in its most basic form. It suggests that in order to solve the shutdown problem, it is necessary to imbue an agent with some sort of information about PREVENT and SELFPRESS. This motivates continuing the study of weaker versions of the shutdown problem.

Downloaded from http://mitp.silverchair.com/isal/proceedings-pdf/isal2024/36/104/2461097/isal_a_00792.pdf by guest on 03 October 2024

References

Armstrong, S. (2010). Utility indifference. Technical report, Citeseer.

Nelson, E. (2023). Incentivizing shutdown by learning to redeploy agents with modified beliefs. *AI Alignment Awards*.

Snyder, M. (2023). The incompatibility of a utility indifference condition with robustly making sane pure bets. *AI Alignment Awards*.

Soares, N., Fallenstein, B., Armstrong, S., and Yudkowsky, E. (2015). Corrigibility. In *Workshops at the Twenty-Ninth AAAI Conference on Artificial Intelligence*.

Thornley, E. (2023a). The shutdown problem: Three theorems.

Thornley, E. (2023b). The shutdown problem: Two theorems, incomplete preferences as a solution. *AI Alignment Awards*.

Downloaded from http://mitp.silverchair.com/isal/proceedings-pdf/isal2024/36/104/2461097/isal_a_00792.pdf by guest on 03 October 2024