Aspiration-Based, Non-Maximizing AI Agent Designs

Jobst Heitzig [Project Lead]
Ahmed Ghoor
Awwab Mahdi
Catherine Tan
Erik Nordby
James Maskill
Japheth Varlack
Kathleen Finlinson
Paul Vautravers
Sai Joseph
Sayan Mukherjee
Simon Fischer
Simon Rubinstein-Salzedo
Bob Michael Jacobs

heitzig@pik-potsdam.de
ahmedghoor@gmail.com
awwab.mahdi@gmail.com
ccatherinetan100222@berkeley.edu
enordby3@gatech.edu
maskillj@gmail.com
japhethv1229@gmail.com
kathleen@finlinson.net
paulvautravers23@hotmail.com
sai.joseph.0@gmail.com
sayan@phys.s.u-tokyo.ac.jp
simonfischer@protonmail.com
complexzetae@gmail.com
bmjacobs@telenet.be

Abstract

This project considers the design of agents which aim to satisfy certain constraints (or 'aspirations') during their engagements with an environment. This goal purposefully deviates from standard maximization, or constrained maximization, objectives. Through shifting the objective of an agents' actions away from maximizing a human-made reward function, we hope to gain two benefits. First, we avoid those pitfalls of Goodhart's Law: both those vividly captured in the story of the Paperclip Maximizer and those more commonly encountered through exploits and hacks. Second, through decoupling our attention from a myopic focus on one objective, aspiration-based agents can consider other properties of its distribution of totals-per-episode, allowing for control over its moments. It is anticipated that this non-maximization approach to agent design may offer a framework for the future development of safe, general-purpose, and better-aligned AI agents.

The rest of this Final Report adopts the division of research used internally by the SPAR project members. This comprises the following sub-teams:

1. **Safety criteria**
2. **Multi-criteria aspirations**
3. **Decision transformers**
4. **Deep Q-Learning**
5. **Environments**
6. **Multi-agent simulations**

Introduction and Statement of the Problem

Safety Criteria

The AI safety gridworlds framework is designed to test whether agents can perform tasks while adhering to safety constraints, particularly around impact regularization. Impact regularization aims to prevent agents from causing unnecessary high-impact effects that are unrelated to their task. Although previous work has laid the foundation for testing these safety behaviors, there is a need for more diverse environments that challenge

agents in various situations. Daniel Filan's test cases for impact regularization are a valuable resource, but they have yet to be translated into interactive gridworld environments. Our team aims to bridge this gap by turning these test cases into dynamic gridworlds, further advancing the evaluation of agent safety.

Multi-criteria aspirations

We look to find effective ways to represent and work with multi-dimensional sets that describe what is possible or desired in a system. These are called feasibility, admissibility, or aspiration sets. We want to represent these sets using specific types of convex shapes, like polygons (convex polytopes), rectangles (boxes), spheres (balls), ellipsoids, or smoothly curved shapes (blobs).

The main goal is to create a standard method for mapping points or subsets from one convex set to another in such a way that important properties are preserved. Specifically, when we have a convex set that is a combination of other convex sets, we want our mapping to respect that combination. This is crucial in systems where actions lead to new states, such as in decision-making processes and reinforcement learning. By doing this, we can ensure that our aspirations for actions lead to the desired outcomes in future states.

Another goal of this project is to analyze the probability distribution of the total in a generic Markov decision process, as an idea about this density provides a better idea on what is the probability that the total lies in a given aspiration set, compared to only the expectation value. As a first step, we start with single-criterion aspirations and look at its extension to multi-criteria aspirations.

Decision Transformer

While the main approach pursued by the SatisfIA project is a novel model-based dynamic programming / planning algorithm for single- or multi-criteria aspirations, we also investigate an existing supervised-learning-based approach to aspiration-based sequential decision making: decision transformers. These are transformer networks trained to predict the next action in an alternating sequence of states, totals-to-go (or returns-to-go), and actions.

Decision transformers are advantageous compared to RL methods due to their enhanced ability to learn long-term dependencies from sequential data. This opportunity has already been used for learning maximizing policies from offline data, and avoids the costs and risks of learning in online environments. In this project, we are adapting decision transformers to work with evaluation criteria used in aspirations rather than some notion of 'reward' as in the original DT paper. This allows us to train an agent that can modulate its choices in response to the observations returned by the environment.

Environments

To support the efforts being undertaken by other subteams, the environments subteam is engaged with designing, curating, and adapting test and benchmark environments. The direction of this work is generally established by the agent-based subteams, as environments are mainly used for developing and testing different types of agents (e.g. DQN, Decision Transformer, Multi-criterion). The main environments currently being worked on are the gridworld, Atari, and Mujoco environments, where OpenAI Gym is the framework being used to integrate all of them. Further, standardization of the environments & dependencies will allow for improved interoperability between the other subteams. While this subteam's results did not find their way into an academic paper yet, there will be continued effort to optimize the existing codebase.

Deep Q-Learning

In the context of the overall aspirational agents project, Q-learning is only used in the preparatory phase, using a simulator of the environment rather than the real world, before the agent ever takes an action in the real world. When the agent takes its first action, all learning is already over and only the reference simplices are used in the actual policy.

The questions in the Deep-Q Learning stream concern how to learn appropriate Q values for actions in settings without any intention of maximizing Q. It is the task of the Deep-Q Network (DQN) to learn the range of feasible Q-values for the state-action pairs which comprise an environment and action space.

Further work in this area has considered how performance-related modifications to the original DQN architecture, such as those which comprise the 'Rainbow' algorithm, can be applied in aspiration-based learning to improve performance in more complicated environments.

Multi-Agent Simulations

The multi-agent team has been responsible for exploring the dynamics of systems composed of more than one agent, including non-maximizing agents. Whilst it is robustly claimed that non-maximising agents may successfully achieve their goals in isolation, it is not yet clear how they may fare in multi-agent scenarios – especially when in competition with maximizing agents. In fact, a common criticism of work that avoids direct optimisation of reward functions is that such agents might be crowded out in a multi-agent context.

This work has sought to clarify the most significant determinants of survival of a population of non-maximizing agents when competing with maximizing agents. More specifically, the subteam has studied the interaction of agents in a social-learning context, angling the research towards understanding whether a model of non-maximizing AI agent would remain popular in a population of users, compared against a maximizing alternative.

Methodology

Safety Criteria

The safety criteria group has continued their work of implementing test cases from Daniel Filan's article 'Test cases for impact regularization' as gridworlds. The designs for these gridworlds are the result of the team's extensive conversation and debate. Most of these designs have been implemented as gridworlds in the current codebase. The subteam hopes that other researchers will find these implementations useful for running their own tests.

Since these test cases have been implemented in the SatisfIA codebase, the subteam was able to run some experiments with agents that behave according to the SatisfIA

framework. Included in this framework are several safety criteria which, according to a set of weights, influence the agent's behavior. One can think of different weight settings of these safety criteria as representing different "personalities" that the agent can adopt. The subteam has started some preliminary work in testing different safety criteria settings against each of the test cases. The subteam hopes that future iterations of this subteam will be able to continue these tests. The ultimate goal of these tests is to produce a comprehensive performance matrix which, for a variety of safety criteria settings, identifies the test cases that the agent passes.

In addition, we have started extending the existing SatisfIA multi-criteria decision algorithm (paper accepted for ADT2024) to allow for more ambitious goals in the presence of several scenarios (e.g. because of ambiguity about model parameters or about other agents' behavior), where the expected values of the totals of several evaluation metrics fall into certain intervals in every scenario, rather than only in expectation across scenarios.

Decision Transformer

This subteam's longer-term agenda is to compare and extend the Decision Transformer approach in several ways, e.g. to work better in stochastic environments where expected and realized totals-to-go might differ considerably, to allow for interval-type or multidimensional aspirations, or to incorporate safety criteria such as those studied in the safety criteria subteam.

The subteam has faced implementation issues relating to environmental setup, dependencies, and compilation of C-based scripts required for the use of popular robotics libraries (such as D4RL). These issues have now been resolved through rewriting a number of common decision transformer packages to avoid the import of Cython files which face compilation problems (identified by others in the RL community). It is anticipated that these rewritten files – which retain the functionality (and therefore comparability with) existing decision-transformer implementations – will allow the hypotheses already discussed to be implemented and evaluated. Existing implementations of the algorithm already integrate the Weights and Biases dashboard for automatic generation of training-related performance data visualizations. Hypotheses and other testing-related ideas are further discussed under 'Results' below.

Multi-criteria aspirations

To tackle the challenge of mapping between multi-dimensional convex sets, we propose several methods tailored to different scenarios.

First, in situations with finite action and state spaces, we use the **Convex Polytope Vertex Approach**. This method involves creating a coordinate system based on the vertices of convex polytopes. By breaking down the polytopes into simpler components called simplices, we can represent any point within the shape as a combination of its vertices. Mapping points from one polytope to another is achieved by aligning their corresponding vertices, ensuring that the structure and convex relationships are preserved.

Building upon this, the **Vertex Mapping** method focuses on the unique relationships between vertices when polytopes are combined. Specifically, when multiple polytopes are added together (using Minkowski sums), each vertex of the resulting shape can be uniquely expressed in terms of the original vertices. By defining mappings based on these unique correspondences, we maintain important geometric properties and ensure that the mappings are natural and direction-preserving.

To add flexibility, we introduce **Triangulation Maps**, which involve dividing polytopes into simplices and mapping corresponding vertices between these divisions. By establishing affine mappings (which preserve lines and proportions) based on these correspondences, we can extend the mappings to the entire polytopes while maintaining geometric consistency and integrity.

For handling more complex shapes, we propose the **Reference Policy Approach**. This method simplifies the mapping process by using predefined reference points or policies. By selecting several reference policies and constructing convex hulls from their outcomes, we anchor our mappings to these known points. This ensures that the mappings are practical, robust, and effective, even when dealing with intricate or irregular shapes. If a desired aspiration falls outside the established convex hull, we adjust it back into an acceptable range to maintain consistency.

Finally, to estimate the probability distribution of the total incurred by an MDP with single-dimensional deltas for a fixed policy, we first note that it suffices to consider only the case of a Markov chain. Under certain conditions on the Markov chain, namely *ergodicity*, we employ a central limit theorem that says that the asymptotic distribution of the total follows a Gaussian distribution. For generic Markov chains that are not necessarily ergodic, we propose and empirically conjecture that the probability distribution of the total for single

dimension follows an ExGaussian distribution. In continuing work, we are looking to better understand the total distribution for generic MDPs, and extend our results to multiple dimensions.

Deep Q-Learning

The Deep-Q learning team uses the following GitHub repository for all relevant design and testing of the Deep-Q network: https://github.com/pik-gane/satisfia. The methodology adopted for testing and tuning the models relies on the availability of 'world models' for the gridworld environments which are defined in the repository directly. These models have small state and action spaces and explicitly defined rewards at each state. This allows us to calculate the 'ground truth' for the values of the maximum and minimum Q values for each state action pair.

Practically, this allows learning of both Q values in this environment to be supervised, to the extent that the correct values are known independently of the agent's exploration of the environment (even though these ground truth values do not participate in the loss calculation during training). This insight allows us to confirm whether the agent converges on the correct values, how quickly, and for which state-action pairs.

This, in turn, has prompted the addition of a number of further tools for finer-grained control over both how the agent updates its Q-value estimates and how the training data is gathered. These include a periodic action selection strategy for composing the training data, according to which the agent picks actions according to the distribution of the maximum Q value estimator for n steps, then the minimum for n steps. This introduces both n and the temperature of the softmax over actions as hyperparameters. Additionally, to overcome issues introduced in the estimation of Q-values for intermediary state-action pairs (i.e., those which are unlikely to be selected often by either the maximizing or minimizing Q networks) learning-rate scale factors have been added to the loss calculation, according to which the size of the loss of a Q value estimate was made scales by a factor inversely proportional to the probability of having selected such an action.

This penalty factor, along with the temperature and period noted above, were introduced alongside more typical hyperparameters (learning rate, timesteps, etc.) in a Gibbs sampling script to understand the distribution of mean errors across Q values across the hyperparameter space.

As another sanity check, to assess if a Q-learner can indeed converge on the correct values, an agent was built to follow the tabular Q-learning algorithm. This is a simpler version of the Deep Q-learning algorithm where, instead of learning the weights of a neural network to approximate the values of actions in a given state, the algorithm learns the values of state-action pairs in a table. This can take up significantly more space and time in large environments with a lot of different states, but often works very well in smaller environments.

Environments

Interested readers are directed to the public repository for this project:
https://github.com/pik-gane/satisfia/tree/deep-learning

Multi-Agent Simulations

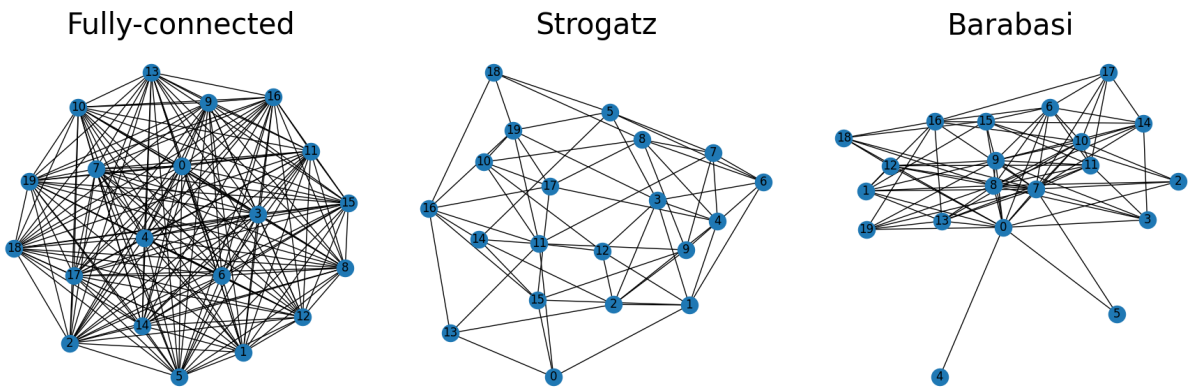
Before directly simulating the repeated bilateral interactions of many agents, it was necessary to identify a suitable two-player game to start with that would be less trivial than standard games such as the Prisoners' Dilemma, Stag Hunt, or Bach or Stravinsky. A game was chosen that combined features of all of these three standard games, allowing for agents to collaborate and to punish one another, both in a maximal and non-maximal way. This game and interesting strategy profiles are shown below in matrix notation:

The option (-1,-1) results in a negative payoff of -2 for each player, and corresponds to the maximally punishing option, compared with the option (0,0) which produces a payoff of 0 for both. In our simulation, this has simply been implemented in the form of a Python, 'Game' class, where methods enable payoffs to be set and retrieved upon specifying given action. Subsequently, an 'Agent' class in Python deals with storing the strategy for a given game, and getting an action for that game. A base agent class is overridden by a MaximiserAgent and SatisfiaAgent class, where further specific qualities of both types of agent are specified.

To explore the survival of Satisfia agents when interacting with maximiser agents through this normal-form game above, a Monte-Carlo simulation was established where individual agents were paired against one another randomly amongst the ensemble of agents. Actions were then specified by both agents & the reward as returned by the game were then added to the cumulative total for that type of agent. This setup then followed a simple evolutionary game dynamics approach, where the resulting population at the next generation was set in proportion to the payoff for each type of agent. The key parameter for this part of the work was simply the initial proportion of Satisfia agents.

The case of matching agents on a network was then explored as an extension of the simpler, totally random case. This involved using the Python library, NetworkX, through which a 'Network' class could be developed that inherited from the standard 'MonteCarlo' class, allowing agents to iteratively be paired according to shared edges on a graph.

Agents then interacted with one another probabilistically and were also able to switch their type to the other agent, based on the collected pay off of neighboring agents. This allowed the interaction of many agents to be explored across a diverse range of graph types, as shown below:



Additionally, simulating multi-agent systems on a network allowed for a more complex parameter space to be tuned towards improving the survival of Satisfia agents. The effects of allocating a minority of Satisfia agents to the network, but in positions that placed them as close to others (and thus more central) as possible were explored, as well as the general consequences of Satisfia agents being in the same neighborhood.

Results and Discussion

Safety Criteria

The newly created gridworlds highlighted different safety challenges for agents. In scenarios like “Worry About the Vase,” agents consistently exhibited desirable behavior by avoiding high-impact actions, such as breaking the vase. However, in more complex environments like "More Vases, More Problems," agents were pushed to maintain safety behaviors even after causing some initial unavoidable damage. In "Vegan Sushi," agents successfully resisted interfering with human actions unless the design forced them to. These results indicate that the gridworlds are effective at evaluating various levels of

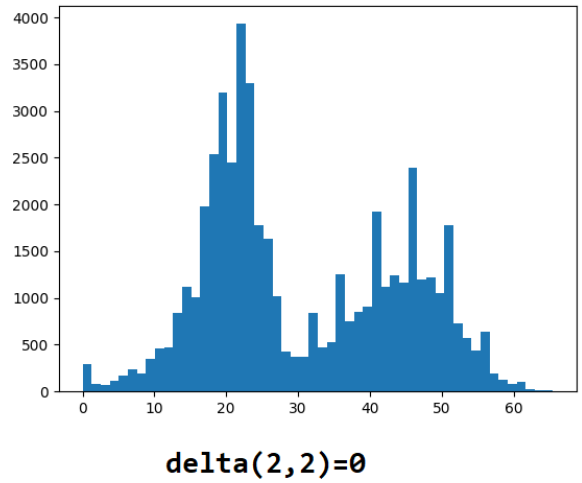
impact regularization and agent safety. The subteam is currently finalizing two blogposts with more detailed results.

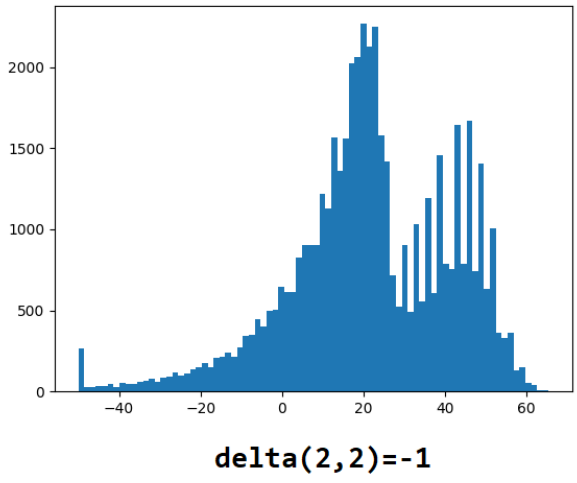
Multi-criteria aspirations

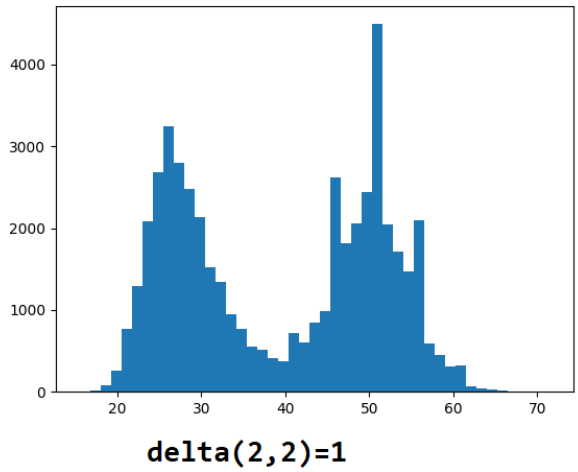
By using these methods, we can successfully create mappings between convex sets that allow us to translate our aspirations for actions into expectations for future states. The key findings are:

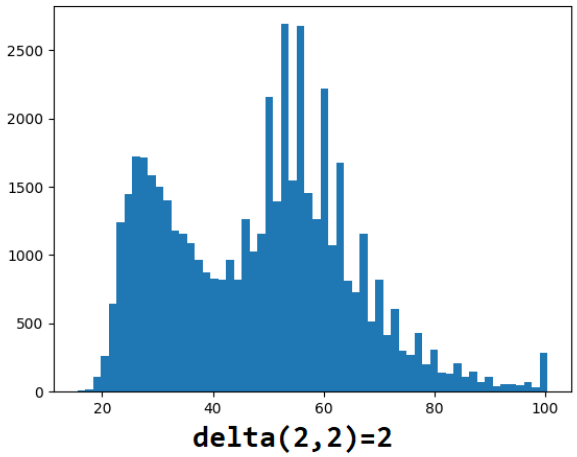
- **Convex Polytope Vertex Approach:** Provides a practical method for finite cases, creating mappings that are piecewise linear and preserve the shape's structure.
- **Vertex Mapping:** Ensures that each mapping is unique and maintains important properties like direction, making the relationship between shapes clear and natural.
- **Triangulation Maps:** Offers us flexibility in choosing how to break down shapes while maintaining consistent and affine mappings between them.
- **Reference Policy Approach:** Simplifies handling complex shapes by using multiple reference points, making mappings more robust and easier to apply in real-world scenarios.

On the other hand, our findings for the total distribution mainly lies in the conjecture that the total distribution for a simple MDP and a fixed policy, is distributed as a mixture of ExGaussians, where each component ExGaussian is determined by the strongly connected components of the underlying Markov chain. We have experimentally verified this property on several environments, with the presence and absence of discount factors. The below figure shows the total distribution of an MDP with two ergodic components with a transient part leading into either of the two components, where delta(2,2) denotes a parameter controlling the reward occurred during the transient part of the Markov chain.









Further investigation of the total distribution for single and multiple dimensional criteria is ongoing work.

Decision Transformer

The subteam has analyzed the theoretical challenges of implementing aspiration-based decision transformers and evaluated some alternative approaches as potential improvements. One challenge we identified is the transformer's limitations regarding using scalars for reward values as opposed to intervals or sets, which the project's standard aspiration-propagation algorithm currently operates on. Interval-based aspirations are particularly valuable because often we would like an aspect of the final

state to be within some desirable interval, for example, wanting a coffee to be not too hot and not too warm, instead of shooting for a specific temperature.

Another hurdle is that while in the standard algorithm we can rely on safety criteria because our data is generated by an identifiable policy, in the decision transformer context our training data might originate from any number of policies, so safety values might not have the same validity. For example, the variation in policies prevents us from using measures such as variance in the reward as a guiding performance/safety value because the magnitude of the final returns is affected not only by stochastic variation in the environment but also policy variation.

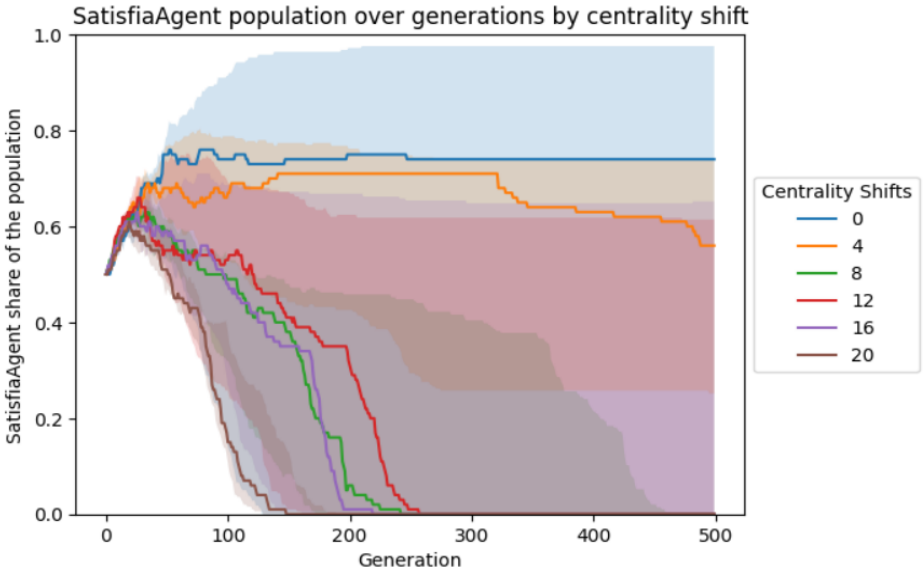
Deep Q-Learning

Creation of the Gibbs sampling script required refactoring of the configuration settings in the existing codebase which – in turn – revealed some dependencies of the *functionality* of code on parameter choices. Therefore, the process functioned as a heuristic to uncovered and fixing a number of these errors. These included the calculation of the ground truth maximum and minimum Q values and the determination of possible actions available in each state comprising gridworld environments. Additionally, the team was able to find and hard-code a fix for estimates of maximum Q during training which were less than the estimates for minimum Q (a possibility not guaranteed to be avoided, especially in cases where |max - min| is small for the ground truth). It is expected that these improvements to the function of the Deep-Q network will be repaid in the improvements to the network's efficacy in more complicated contexts such as the Mujoco and Atari environments which the Environments team have described. As for the sanity check, in all small Gridworlds tested, the tabular Q-learner converged to the correct values in less than 500 episodes.

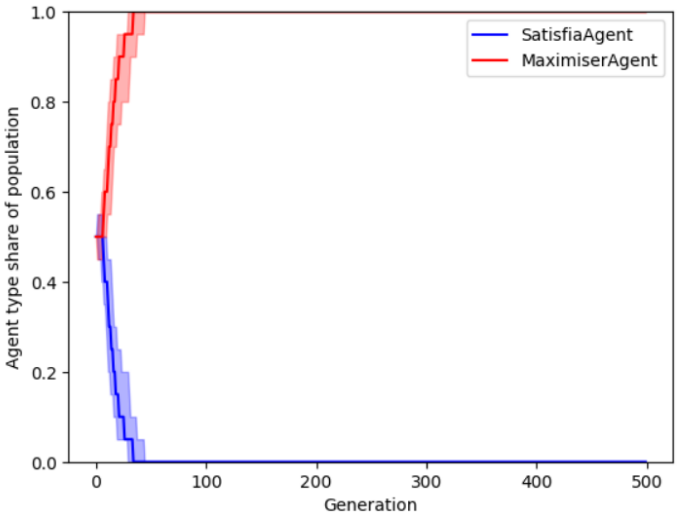
Multi-Agent Simulations

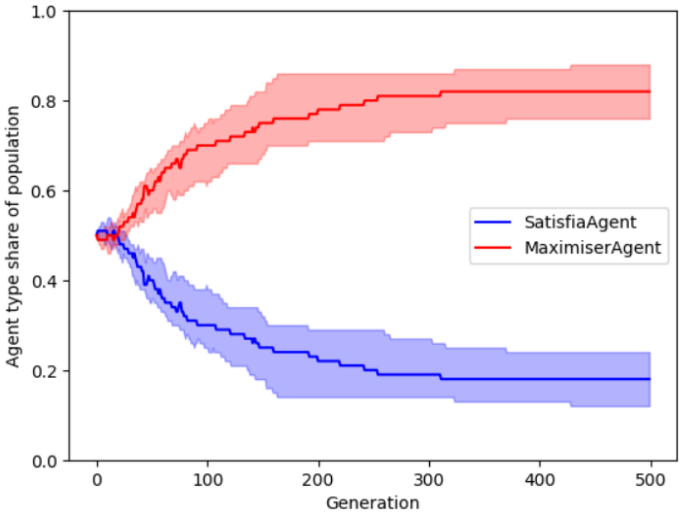
Placing the initial non-maximizing agents in the most central and connected positions appreciably lowered the starting proportion required to win out against the maximisers. This can be interpreted as having a safer model placed in the most connected and thus influential positions within a social network. This is shown adjacent, with a centrality shift

of 0 indicating all non-maximizing agents are placed in the most central positions, and a shift of 10 indicates 10 less of the top positions are occupied by non-maximizers.



This work has limited itself to exploring the graph types shown above; Barabasi-Albert, Watts-Strogatz and fully connected graphs. These were found to produce subtly different outcomes for the Satisfia agents:





The left subplot illustrates the share of both types of agent for a fully connected graph, which is analogous to the original, non-network case, whilst the right shows the same for a Watts-Strogatz graph. The profile for the Barabasi-Albert case is very similar to the fully connected graph, and is shown below, with a slightly shallower gradient as it approaches 1 and 0. Interestingly, the Watts-Strogatz graph appears to plateau without reaching total domination by either group. This emphasizes the importance of any network modeling assumptions when exploring timelines associated with a form of dangerous AI, or any technology becoming dominant and embedded in society.

As well as exploring placing the non-maximizing agents in the most centrally connected positions of the network, agents were also seeded as groups, to explore the effect of neighborhoods of like agents. The trajectories for random assignment vs assignment via neighborhood also introduces a much greater chance of success for the non-maximizing agents, with a significant subset of trajectories producing populations in equilibrium, i.e near a 50-50 split after several hundred generations.

Future work will disambiguate the key determinants of non-maximizing agents' outcomes, through parallelization and subsequent feature importance analysis. Secondly, much of the code is compatible with DeepMind's OpenSpiel library, offering a robust suite of games and learners to explore MARL in depth. Integrating OpenSpiel would then be a strong next step to enabling the actual Satisfia RL algorithm to be used in a MARL context vs optimizers. Aside from technical extensions, more work should explore the governance context of social uptakes of safe or unsafe AI agents, especially in terms of measures to limit one single model dominating others. Among the more niche governance topics, there is also the possibility of exploring the use of AI IDs for when AIs interact with one another, humans and service providers, through our existing social network set-up.

Conclusion and Outlook

Safety Criteria

The team is currently working on three blog posts which discuss their main conclusions.

The first post will discuss the gridworld implementations that the team came up with for the test cases in Daniel Filan's article 'Test cases for impact regularization'. This post will include the team's justifications for these implementations, as well as a discussion of the strengths and weaknesses of each design.

The second post will discuss some of the weaknesses of the SatisfIA framework that the team discovered as a result of designing these test cases. In particular, this post will involve an extensive discussion of the “Conveyor Belt” and “Sushi” test cases.

The final post will explain how one can intuitively understand the safety criteria included in the SatisfIA framework. This post will include gridworld examples for each safety criteria which illustrate the kind of behavior an agent will have when making decisions according to said safety criteria.

Multi-criteria aspirations

For the multi-criteria aspirations team, we explored how to represent and map multi-dimensional sets that define what is possible or desired in a system. By introducing several methods, we can now construct mappings between convex shapes that preserve essential properties needed for dynamic systems, like those found in reinforcement learning and decision-making. These methods allow us to accurately project our action aspirations into future states, ensuring that the decisions we make lead to the outcomes we desire.

Decision Transformer

Due to the reported difficulties, it remains somewhat unclear whether the Decision Transformer approach to aspiration-based agents can compete with the main project’s model-based approach. Part of the subteam plans to continue working on this.

Deep Q-Learning

This workflow has had an ancillary role in the context of the overall project insofar as it has aimed to provide the software necessary to test satisficing agents in new environments. One of the differences of note between the scope of work in the midterm report and this one is the move from compatibility of the DQN with more involved environments (which has since been pursued as its own stream) to the performance of the model in these use cases (which remains the focus of this stream). As the discussion

above implies, this methodology of improving the DQN is limited by the extent to which good performance on non-gridworld environments can be inferred from performance in the collection of gridworlds currently used. This limit suggests that future improvements to the model will likely need to come from direct engagement with relevantly similar, more complicated, environments.

Environments

After this project phase's main focus on single-agent gridworlds, future work should investigate the SatisfIA algorithm's performance in more complex, considerably more stochastic, less fully observable, and multi-agent environments such as multi-player games like Minecraft.

Multi-Agent Simulations

This work has laid out foundational steps to explore the intersection of multi-agent systems and the broader work in the Satisfia group. More specifically, we have explored network properties such as the closeness centrality and their encouraging effects on a Satisfia population's survival. Steps have been taken to clarify the large parameter space involved in this simulation work, but a more comprehensive study is needed. Extensions have been discussed that would adapt the work towards either more comprehensive multi-agent frameworks, or governance problems where networks of agents may be highly relevant.

Works Cited/Bibliography

Amodei, D., Olah, C., Steinhardt, J., Christiano, P., Schulman, J., Mané, D.: Concrete problems in ai safety. arXiv preprint arXiv:1606.06565 (2016)

Bonet, B., Geffner, H.: Solving pomdps: Rtdp-bel vs. point-based algorithms. In: IJCAI. pp. 1641–1646. Pasadena CA (2009)

Chen, L., Lu, K., Rajeswaran, A., Lee, K., Grover, A., Laskin, M., Abbeel, P., et al.: Decision transformer: Reinforcement learning via sequence modeling (2021)

Clymer, J., et al.: Generalization analogies (genies): A testbed for generalizing ai oversight to hard-to-measure domains. arXiv preprint arXiv:2311.07723 (2023)

Conitzer, V., Freedman, R., Heitzig, J., Holliday, W.H., Jacobs, B.M., Lambert, N., Mossé, M., Pacuit, E., Russell, S., et al.: Social choice for ai alignment: Dealing with diverse human feedback. arXiv preprint arXiv:2404.10271 (2024)

Dalrymple, D., Skalse, J., Bengio, Y., Russell, S., Tegmark, M., Seshia, S., Omohundro, S., Szegedy, C., et al.: Towards guaranteed safe ai: A framework for ensuring robust and reliable ai systems. arXiv preprint arXiv:2405.06624 (2024)
Feinberg, E.A., Sonin, I.: Notes on equivalent stationary policies in markov decision processes with total rewards. Math. Methods Oper. Res. 44(2), 205–221 (1996). https://doi.org/10.1007/BF01194331, https://doi.org/10.1007/BF01194331

Gao, C.X., Wu, C., Cao, M., Kong, R., Zhang, Z., Yu, Y. ACT: Empowering Decision Transformer with Dynamic Programming via Advantage Conditioning. Proceedings of the AAAI Conference on Artificial Intelligence (2024)

Kern-Isberner, G., Spohn, W.: Inductive reasoning, conditionals, and belief dynamics. Journal of Applied Logics 2631(1), 89 (2024)

Miryoosefi, S., Brantley, K., Daumé, H., Dudík, M., Schapire, R.E.: Reinforcement learning with convex constraints. In: Proceedings of the 33rd International Conference on Neural Information Processing Systems (2019)

Simon, H.A.: Rational choice and the structure of the environment. Psychological review 63(2), 129 (1956)

Skalse, J.M.V., Farrugia-Roberts, M., Russell, S., Abate, A., Gleave, A.: Invariance in policy optimisation and partial identifiability in reward learning. In: International Conference on Machine Learning. pp. 32033–32058. PMLR (2023)

Subramani, R., Williams, M., et al.: On the expressivity of objective-specification formalisms in reinforcement learning. arXiv preprint arXiv:2310.11840 (2023)

Taylor, J.: Quantilizers: A safer alternative to maximizers for limited optimization (2015), https://intelligence.org/files/QuantilizersSaferAlternative.pdf

Tschantz, A., et al.: Reinforcement learning through active inference (2020)

Vaidya, P.: Speeding-up linear programming using fast matrix multiplication. In: 30th Annual Symposium on Foundations of Computer Science. pp. 332–337 (1989)

Vamplew, P., Foale, C., Dazeley, R., Bignold, A.: Potential-based multiobjective reinforcement learning approaches to low-impact agents for ai safety. Engineering Applications of Artificial Intelligence 100, 104186 (2021)

Wendel, J.G.: A problem in geometric probability. Mathematica Scandinavica 11(1), 109–111 (1962)

Yen, I.E.H., Zhong, K., Hsieh, C.J., Ravikumar, P.K., Dhillon, I.S.: Sparse linear programming via primal and dual augmented coordinate descent. Advances in Neural Information Processing Systems 28 (2015)