The Effects of Heuristic Optimisations on Planning Algorithms Within Cooperative AI

Cooperative Planning in Overcooked

Jonte Herben[1]

Supervisor(s): Frans Oliehoek[1]**, Robert Loftin**[1]

[1]EEMCS, Delft University of Technology, The Netherlands

A Thesis Submitted to EEMCS Faculty Delft University of Technology,
In Partial Fulfilment of the Requirements
For the Bachelor of Computer Science and Engineering
June 25, 2023

Name of the student: Jonte Herben
Final project course: CSE3000 Research Project
Thesis committee: Frans Oliehoek, Robert Loftin, Klaus Hildebrandt

Abstract

Cooperative AI is AI designed to cooperate with humans. One example of such an AI, made using planning algorithms, was studied in a paper from 2019 which used a simplified version of the video game Overcooked for evaluation. However, only limited evaluations were possible due to the long runtime and heuristic optimisations made. This paper will attempt to increase the amount of evaluations while removing an important optimisation limiting the functionality of the AI: the omission of counters. In the end it will find ways of reducing the runtime and increasing performance, which together allow for the addition of counters in a few instances. The final results with counters suggest there is more to gain upon removal of a select few heuristic optimisations and improvement of the human behavioural clone used to simulate a human.

1 Introduction

Cooperative AI is a field dealing with AI specifically made to work with humans[1]. The main difficulty in this field is not so much creating an optimal AI, instead it is creating one that can handle a suboptimal human. One way of doing so is through machine learning, as it can learn to work around the complex behaviours of a human. However, most machine learning methods offer little transparency and require human data to be trained on. This paper will focus on a version of cooperative AI that does not suffer from these problems: planning algorithms.

Planning algorithms could be described as "a search for a sequence of logical operators or actions that transform an initial world state into a desired goal state."[2] Or, in other words, an algorithm to create a plan to get from state A to state B. This search usually involves an exploration of the problem space, which can lead to a significant runtime in complex environments.

In 2019 a paper was written by Carroll et al. titled "On the Utility of Learning about Humans for Human-AI Coordination"[3], which compared multiple methods of creating a cooperative AI, both machine learning and planning algorithms. The environment used for evaluation was a simplified version of the video game Overcooked, a game that gives players a score based on how well they cooperate. In the paper they found optimal planning algorithms to be too slow to be feasibly evaluated, so they had to make multiple heuristic optimisations, leading to, as described in the paper, near optimal algorithms instead[3]. This paper will analyse the impact of one such optimisation made, namely the omission of counters.

In short, this paper aims to answer the question: "How can cooperative planning within Overcooked be improved by removing heuristic optimisations?" On top of this it will recreate the results from the original paper by answering how planning can be used to achieve cooperative AI. The specific heuristic sub-optimisations, also analysed, will answer what the impact of more focused heuristics and incorporating history is on cooperative planning.

The answer to these questions will be able to aid future research in deciding when to use what kinds of heuristic optimisations. It will also directly advise on how to further improve the planning agents in the Overcooked-AI environment.

The final conclusion reached in this paper suggests that more focused heuristics and history achieve their intended results of lowering the runtime and improving the score, respectively. The final planning agents using counters still take too long to be evaluated in many situations, and achieve low scores when they are. This is due to the low priority of counters and the limitations of the human behavioural clone.

The paper will answer the research question according to the following structure: first it will discuss the preliminaries to this research in section 2. It will then go on to explain the step-by-step plan to reach a conclusion in section 3, and the specific contributions to the planning algorithms made in this research in section 4. section 5 will then detail the final results, which will be analysed in section 6. Following this will be an ethical consideration of this research in section 7. Finally, the conclusions will be summarised together with possible future work in section 8.

2 Preliminaries

This section will explain the work and data used in this research. First, it will explain the evaluation environment in subsection 2.1, followed by the paper this research is built upon in subsection 2.2.

2.1 Overcooked

Overcooked[1] is a video game where two or more players are tasked with preparing meals by combining or processing ingredients according to certain recipes. The gameplay heavily revolves around cooperation with other players. Recipes often require multiple steps, to be completed separately from each other, incentivising cooperation. On top of this, players are not able to move through each other, which creates further dependencies on other players. This collision mechanic is often focused on in the layout of the levels, giving the players very little room to move around, with some even going as far as to restrict players to their own separate sections of the kitchen, forcing them to pass ingredients over counters to make a delivery.

This dependency on cooperation, tied to a score, is what makes it an ideal environment to test cooperative AI. However, inserting the cooperative agents straight into the game comes with unneeded complexity, in both the visuals and level layouts. Instead, it is easier to create a simpler version made to run as fast as possible, while also allowing for full control over the evaluation environment. For that reason, the paper by Carroll et al.[3] created a simplified version of the Overcooked, visible in Figure 1. This environment was used as the foundation for the research conducted in this paper. The environment includes only a single recipe: onion soup, prepared by putting three onions into a pot, turning it on and

[1] https://www.team17.com/games/overcooked/

waiting for it to cook, before collecting it with a plate and delivering it. Each layout consists of: walkable tiles, counters to put items down on, pots, onions, and delivery tiles.

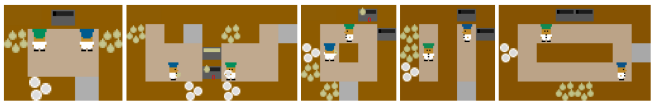


Figure 1: The 5 layouts in the simplified version of Overcooked, made and used in the paper by Carroll et al.[3] From left to right: Cramped Room, Asymmetric Advantages, Coordination Ring, Forced Coordination, and Counter Circuit.

2.2 On the Utility of Learning about Humans for Human-AI Coordination

The two planning algorithms compared in the paper by Carroll et al.[3] are 'coupled planning with replanning' and 'model-based planning'. Both of these algorithms use hierarchical A*, a version of the A* pathfinding algorithm that splits up the problem space into a higher- and lower-level, called medium and low level actions in the original paper, to optimise traversal.

The environment used in the paper, as well as all algorithms and models, are publicly available as a Github repository[2]. This repository, the overcooked environment, the planning algorithm implementations and the behavioural cloning models were used as a foundation for the research presented in this paper.

Coupled planning with replanning

Coupled planning computes the optimal path for both agents and plays their part in that plan, essentially assuming the other player will play optimally as well. To account for the human deviating from the planned path, the agent replans their plan after every timestep.

Model-based planning

Model-based planning uses a model of the other player to predict its actions. It then plans an optimal path around that. The paper uses a machine learning model trained on human player data for its predictions.

Behavioural Cloning

The paper and the repository also include machine learning models for every layout trained to mimic a human. This speeds up evaluations by removing the constant need for human test subjects. It was found in the paper by Carroll et al. that the models would sometimes get stuck by repeating the same action over and over again. To combat this a manual check was added to take a different action if the model was stuck for three timesteps in a row.

Conclusions in the original paper

The relevant conclusions made in the original paper are as follows:

- Coupled Planner - Performs well when paired with itself, but significantly worse when paired with the behavioural clone.
- Model-based planner - The model-based planner was evaluated with both a different behavioural cloning model, as well as the exact one it uses during planning. The latter scenario was intended to measure a what-if scenario where the planner could perfectly predict the other agent's actions. These perfect predictions proved to lead to significantly higher scores than the standard model, proving there is much to gain in increasing the accuracy of the prediction model. The standard planner even scored zero on half of the evaluations as it would get stuck in a loop with the other agent.

3 Removing Heuristic Optimisations

This section will outline the steps taken in answering how cooperative planning in Overcooked can be improved by removing heuristic optimisations. It will start by explaining the steps to reproducing the results from the original paper by Carroll et al. in subsection 3.1, followed by the process of measuring the impact of the omission of counters, explained in subsection 3.2.

3.1 Reproducing Results

In order to measure the impact of the optimisations, the baseline results will be measured first. To this end, the original results from the paper by Carroll et al. will be reproduced through the Github repository containing the codebase used in the paper[2]. This repository will also be used to achieve all other results in this research. This repository contains a file that contains almost all code to run the evaluations needed. The code to evaluate the coupled planning agent with the behavioural clone is missing, and will thus be added.

In the paper by Carroll et al., evaluations of the planning algorithms were only executed for the first two of the five layouts and for only 100 out of an intended 400 timestep. One 'timestep' being an action done by both agents. This evaluation will also be done in order to compare the new results to the original ones.

3.2 Omission of Counters

The current implementations of the planning agents do not consider picking up or dropping off items at counters, due to the drastic amount of options it adds to the search space, which leads to a drastic increase in runtime. The impact this has on the score will be measured by adding counters back in. The existing code has a list of counters for each map, which was set to be empty in order to remove them. Adding them back in will be as simple as updating this list.

However, just adding counters to the search space will lead to a runtime too long to be feasibly evaluated, thus requiring the need for additional heuristic optimisations. The main additional optimisation will be the improvement of the A* prediction heuristics. On top of this, the accuracy of the model inside the model-based planner will be improved by correctly setting its history. These changes, along with some additional smaller optimisations will be explained in detail in section 4.

[2] https://github.com/HumanCompatibleAI/overcooked_ai/releases/tag/neurips2019

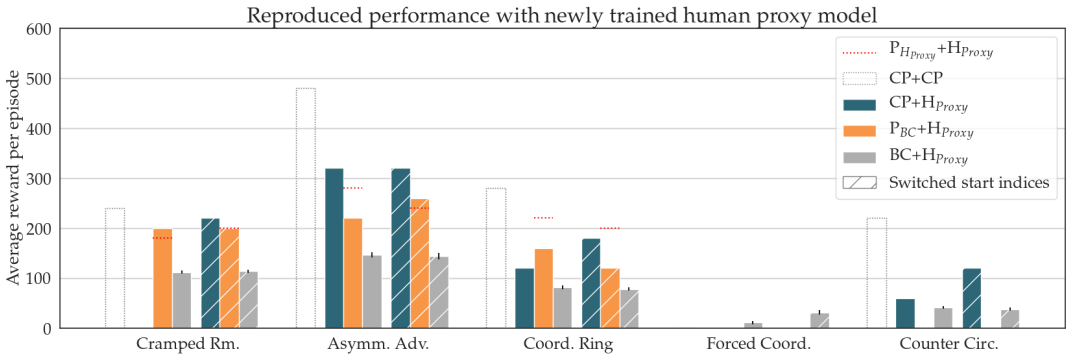


Figure 2: Reproduced results from the paper by Carroll et al. with a retrained behavioural cloning model.[3]

These smaller optimisations will also be evaluated on their own in order to more accurately evaluate their impact.

The final impact of counters will mainly be measured through the Forced Coordination layout, as the optimal path includes the use of counters, unlike three of the other four layouts. The last layout, Counter Circuit, also has an optimal path involving layouts. However, unlike Forced Coordination, it does not force the agents to use them. Therefore the agents will have to look through a lot more states before finding the optimal path, probably increasing the runtime to infeasible levels. This layout will still be attempted to be evaluated, just not a focus of this research.

3.3 Evaluating Runtime

Since a significant portion of the work that will be done during this research will be optimising the runtime of the planning agents, evaluations of those will as be done. In order to evaluate this, not the time but the amount of states explored during the medium-level A* search will be measured. This statistic is independent of the specifications of the system running the code, and is linearly correlated to runtime, as when the planning agent has to look at twice as many states before finding the optimal path, the code will run for twice as long. All changes that will be made will also mainly impact this number. These numbers can, however, not be used to compare between the coupled planner and the model-based planner, as the model-based planner does a lot more work for each state it explores.

4 Specifics of Adding Counters

The option to drop items or pick them up is implemented in the current codebase, but is restricted to a list of valid locations. This list is currently empty, and adding in counters is as simple as changing this list. However, running this as it is takes too long. Because of this, multiple supplementary optimisations were introduced. First the priority of counters were decreased in subsection 4.1, followed by improvements to the A* heuristics in subsection 4.2. Finally, the model inside the model based planner was given an accurate history in subsection 4.3.

4.1 Decreased Priority of Counters

All actions involving counters, picking up and dropping items, were decreased in priority to the same level as waiting. Originally, when considering what medium-level action to do(I.E. picking up an onion, delivering soup), the agent would only consider waiting if no other option is available. When adding the options to pick up or drop items, the agent would consider picking up or dropping an item every state, which proved to be too complex to feasibly evaluate. Therefore both picking up and dropping off items were decreased in priority to be similar to waiting.

This change had multiple side-effects on how the agent interacted with counters. However, in the layouts looked at, it was either the only option(Forced Coordination layout), or only reasonable when nothing else is possible. The only layout it would negatively affect was the Counter Circuit layout, as the usage of counters is optional, but optimal.

A generic change was also made to the coupled planning agent. Originally, both agents could only consider waiting together, meaning as soon as one of them could do nothing but wait, both of their search spaces increased. This was changed to be based on each individual agent, hopefully leading to less states explored overall.

4.2 Improved Heuristics

The heuristic function used to predict the cost of a new state during the medium-level A* search was also improved. During the a* search, the next node to explore is chosen on the basis of the predicted timesteps needed to deliver the soup. Originally, it did not consider distance between the items and their destination, and would see an onion held by a player as equal to one in a pot. The distance between items and their destination was added, as well as the time left to cook soup. These changes were hypothesised to differentiate between more states, and lead to less states being considered.

4.3 Adding History to Model

The behavioural cloning model has protection that makes it take a different action if it is stuck in the same state for 3 timesteps. This is based on a history stored inside the model. However, because the planner uses A*, states are often explored non-linearly, preventing the model inside the model-based planner from keeping an accurate history and getting

itself unstuck. This was fixed by adding the history to each state explored during the A* searches.

The way the planner handles duplicate states was also changed, where before the planner would not explore a state it has already seen, now it counts the times it has seen a state. This is to give the model time to potentially break out of the loop.

5 Results

This section will showcase the obtained results, starting with reproduced results from the paper by Carroll et al. in subsection 5.1. Followed by the results for the new heuristics in subsection 5.2 and the model-based planner with history in subsection 5.3. Finally,the results of the agents using counters will be presented in subsection 5.4. These results can be reproduced from the codebase attached to this paper. That codebase will have a readme detailing the specific instructions needed to reproduce the results shown in this section.

5.1 Reproducing Results

Figure 2 Shows the reproduced results for all layouts and for the full 400 timesteps. For these evaluations the behavioural cloning model was also retrained using the data in the original repository. The most significant difference with the original results is a score of zero for the first layout between the coupled planner(CP) and the behavioural clone(H_{Proxy}), caused by both agents picking up an onion and getting stuck due to their inability to drop them. This behaviour was not present in the results of the original paper, nor mentioned.

The second significant difference is in the reward of the model-based planner. The original paper concludes there is much to gain by improving the internal model of the planning agent. However, the reproduced results show little room for improvement between the rewards of the model-based agent with the training model(P_{BC}) and the model-based agent with the test model($P_{H_{Proxy}}$); the former even exceeding the latter's score in one situation.

Original Behavioural Cloning Model

The significant differences between the original and the reproduced results were theorised to either be the product of a change in the behavioural cloning agent or some uncontrolled randomness in each run. To test these hypotheses, the evaluations were rerun with the original behavioural cloning agent for just the layouts in the original paper. The evaluations were also repeated five times, with the exact same cloning agent and parameters, to showcase the randomness between each run. The results of these evaluations are visible in Figure 3. These results feature the same differences as before; the coupled planning agent gets stuck and achieves a very low score, and the model-based planner shows little reward to gain by improving the model.

Finally, to rule out the possibility of the up-scaling of rewards done in the original paper affecting the results, the results were also evaluated with the original model and a timestep of 100. These results multiplied by four are visible in Figure 4. These results do show room for improvement for the model-based agent on the second layout, but the other differences remain the same.

The planning-based agents were also evaluated for the three layouts not used in the paper due to their complexity. Results of the evaluation are displayed in Figure 2. The Forced Coordination layout only results in scores of zero as the counter optimisation makes it impossible. The Counter Circuit layout is also missing the model-based planning results as the evaluations were estimated to take roughly ten weeks, which is out of scope for this research.

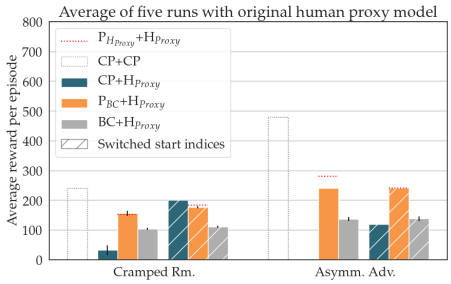


Figure 3: Recreated results using the original behavioural cloning agent, and taking the average of five runs.

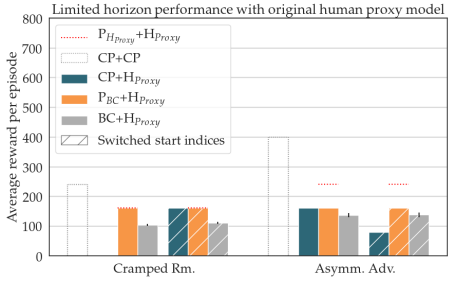


Figure 4: Recreated results using the original behavioural cloning agent. Only evaluated for 100 timesteps and scaled to the full 400 timesteps.

5.2 Improved Heuristics

The results of the improved heuristics are shown in Figure 5a and Figure 5b for the coupled planner, and in Figure 7a and Figure 7b for the model-based planner. The reward achieved with the improved heuristic is not significantly different overall; the agent with the new heuristics reaches a higher score in some cases and lower in others. The main difference presents itself when looking at the performance. The new heuristic leads to less states explored in all cases except for the coupled planner in the Cramped Room layout, which is caused by the original coupled planner getting stuck. This difference is more extreme on more complex layouts, with some even presenting a magnitude of difference between the amount of states explored.

5.3 Model-Based Planner with History

The reward and runtime of the addition of history to the model based planner are visible in both Figure 7a and Figure 7b, respectively. Even though the evaluations were only completed for the first and half of the second layout, a clear trend

(a)

(b)

Figure 5: (a) Rewards for all coupled planning agent variations for all layouts except Forced Coordination. All variations were evaluated on all layouts except for the agent with counters, which was only evaluated on the Cramped Room layout due to time constraints. (b) Average states expanded per timestep for the same agents and layouts as (a). Notice the logarithmic scale, used to properly show the extreme differences between some agents. The amount of states expanded is linearly correlated with the runtime and machine independent, and was therefore used as a more objective measure of the impact on runtime.

is still visible in both graphs. In every evaluated situation, the reward is higher or equal to the standard planner and the runtime is worse. This is expected, as the internal model becomes more accurate, but the planner has to check each duplicate state multiple times to see if the model breaks out of the loop, whereas the normal model rejects every duplicate state.

5.4 Counter Results

In the end, the results for the planning agents with counters were only obtained for the Forced Coordination layout and the coupled planner in the Cramped Room layout. This is due to the runtime still being too high for most layouts.

Cramped Room

The reward for the coupled planning with counters are both better for the regular starting indices, as the agent successfully puts down their onion instead of getting stuck. However, the reward is still roughly half of the results of the coupled planner that do not get stuck. This is due to the counters in the layout slowly filling up with items, which are not picked up because the agent only considers counters once no other action is possible. Once all counters are filled up, the agent breaks in the same way as the planner without counters, as it cannot put its item down.

The reward for the switched indices is almost zero, because the human proxy grabs the soup and gets stuck in a loop in front of the delivery tile. This is probably due to the filled counters creating a state the model did not encounter during training. These counters are filled up because the planning agent frequently drops items while waiting, as it does not see that as a negative thing, since it does not directly impact the amount of timesteps needed to make a delivery.

Forced Coordination

The Forced Coordination layout took too long to evaluate for the regular starting indices of the coupled planning agent, and for the switched indices of the model-based agent. This was due to the suboptimal performance of the human proxy agents, leading to states that required complex plans to achieve a result. The rewards that were achieved were high for two coupled planners playing together, and for the human proxy based planner, although the coupled planners scored much higher than the model-based planner, as it did not involve the poorly performing human proxy.

The reward for the other two results, one of the model-based planner and one of the coupled planner, was zero. In both cases the planning agents became stuck in a loop with the behavioural cloning agents, which they did not break out of.

5.5 Complete Comparison

A complete comparison between the planning agents and the human proxy agents is visible in Table 1. This table only includes results that simulate a planning agent playing with a human, so two coupled planning agents and a model-based agent with the model of the other agent were left out. This

leads to low scores for the agents using counters, as they performed poorly when paired with an agent they could not predict.

6 Discussion

This section will analyse the different results obtained for each subject.

6.1 Recreated Results

The recreated results do not match the original results in the paper by Carroll et al., despite multiple attempts to find an explanation for the randomness. Even with the original behavioural cloning model and technique the coupled planner picks up an onion at the same time as the human clone and gets stuck, and the difference between a fully accurate model-based planner and a regular one is much less than reported.

6.2 Improved Heuristics

The improved heuristic does not result in a significantly different score; for most instances the score is comparable to a model without. Only three instances resulted in a significant difference in reward, but there does not seem to be any pattern between those cases.

The main difference made by the heuristics is the runtime, with significant decreases in expanded states ranging from a slight decrease to almost one-tenth of the regular planner. This difference is most apparent in more complex layouts.

6.3 Model-based Planner with History

Adding history to the model-based planner results in a higher reward in all evaluated cases, although it also increases the runtime significantly. This increased runtime is what limited the situations that were evaluated. This improvement does require more evaluations before it can be claimed to achieve higher scores, but the theory of improving the accuracy of the model inside the model-based planner does suggest it.

6.4 Planners with Counters

The evaluations obtained from the planning agents using counters are limited due to the poor runtime, and it is thus hard to conclude anything about them. The Cramped Room evaluations show the usage of counters allow the planning agent to escape a state where both agents are holding onions while the soup is cooking, albeit temporarily until all counters fill up. Increasing the priority of counters would fix this by allowing the agent to pick the items back up when needed, but would also drastically increase the runtime.

The Forced Coordination evaluations achieve low scores when paired with an agent that does not do exactly as they predict. This is due to the layout, which forces agents to work with one another instead of around one another, unlike the other layouts. In the current evaluation both the planning and behavioural cloning agent do not adapt to each other, leading to a score of zero. In a realistic scenario where a planning agent would be paired with an actual human, the human would realise they are stuck in a loop, and adapt to the planning agent. Therefore, the low scores observed are likely due to limitations of the behavioural clone.

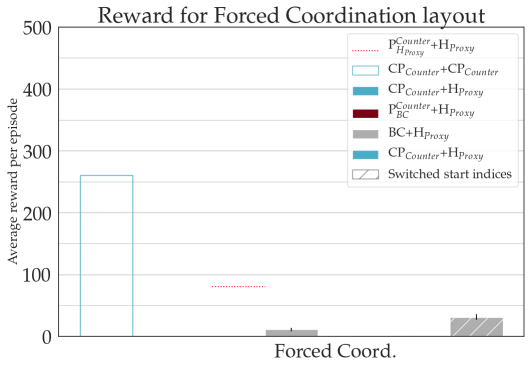


Figure 6: Rewards for the Forced Coordination layout. Only the agents with the counter optimisation are shown as the layout requires the agents to use counters. Three results are missing due to time constraints, namely the coupled planner with the human proxy for the regular starting indices, and the two model-based planning evaluations for the switched start indices. Other results that appear to be missing are instead zero.

7 Responsible Research

This section will discuss the ethical aspects of this research. The main points that will be addressed are the reproducibility of this research and the use of human data.

7.1 Reproducibility

In an effort to make the results achieved in this research as reproducible and transparent as possible, the codebase used will be supplied along with this paper. That codebase will contain a readme and clear instructions on how to generate both the results and the graphs shown in this paper.

7.2 Human Data

In order to train the behavioural cloning models human data had to be used. The data set used was the one present in the repository from the paper by Carroll et al.[3]. This is a data set of human play sessions of the Overcooked-AI environment. This data has already been made anonymous by Carroll et al. and will thus not be a privacy risk.

8 Conclusions and Future Work

This paper aimed to answer the question: ”How can cooperative planning within Overcooked be improved by removing heuristic optimisations?” In order to do so, it would also recreate results from the original paper by Carroll et al. and measure the impact of improving A* heuristics and incorporating history into planning, two optimisations necessary to run the heuristic optimisation analysed: the omission of counters. This section will discuss what answers were found and where to go from here.

The recreated results were found to be significantly different from the original ones, even after using the same limited horizon evaluation technique and the same evaluations were executed multiple times to account for randomness. Where the original paper showed both the coupled planner and model-based planner as leaving room for improvement; the newly obtained results showed the opposite, apart

Table 1: Rewards for every planning agent and optimisation when paired with the human proxy. The highest reward for each layout and starting indices have been marked in **bold**. The average reward for all layouts is also shown, where missing results were treated as zero.

Layout	Start	CP	CP_{Heur}	$CP_{Counter}$	P_{BC}	P_{BC}^{Heur}	P_{BC}^{Hist}	$P_{BC}^{Counter}$
Cramped Rm.	$AI + H_{Proxy}$	0	180	100	**200**	**200**	**200**	-
	$H_{Proxy} + AI$	**220**	180	20	200	200	**220**	-
Asymm. Adv.	$AI + H_{Proxy}$	**320**	**320**	-	220	**320**	-	-
	$H_{Proxy} + AI$	320	**340**	-	260	220	-	-
Coord. Ring	$AI + H_{Proxy}$	120	**180**	-	160	160	-	-
	$H_{Proxy} + AI$	180	180	-	120	**200**	-	-
Forced Coord.	$AI + H_{Proxy}$	0	0	-	0	0	0	0
	$H_{Proxy} + AI$	0	0	0	0	0	0	-
Counter Circ.	$AI + H_{Proxy}$	60	**80**	-	-	-	-	-
	$H_{Proxy} + AI$	**120**	20	-	-	-	-	-
Average		134	**148**	12	116	130	42	0

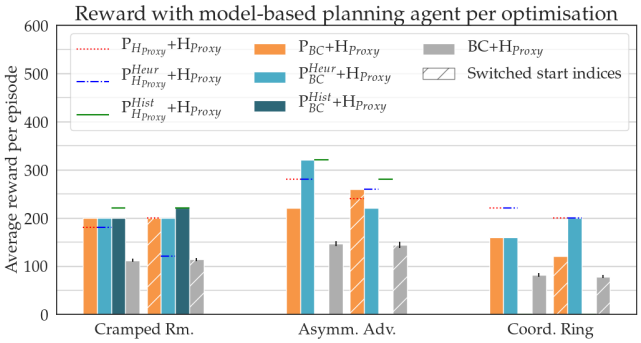
from one instance where the coupled planner got into an inescapable situation and achieved a near zero score, which was also not present in the original results.

Improving the heuristic predictions used in the A* search of both agents by accounting for distance between an item and its destination led to no significant difference in score, but a drastic decrease in runtime, especially for the more complex layouts.

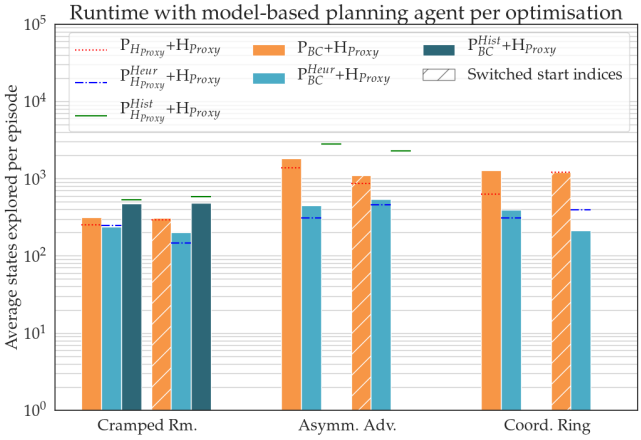
Adding history to the models inside the model-based planner improved their accuracy, and increased all scores measured. However, doing so also increased the runtime of the evaluations, and thus only a few situations were evaluated. This improvement could benefit from more tests to allow for a more concrete conclusion.

With both of these improvements the planning agents with counters ran in reasonable time for half of the situations of the Cramped Room and Forced Coordination layout. The first layout caused the coupled planner to get stuck because it could not put away its item. This was fixed by the introduction of counters, but only temporarily, due to an optimisation decreasing the priority a planning agent gives to picking up or dropping off items. Further improvement to this could be made by removing this optimisation, although this would need to be replaced by a different optimisation, as it helped keep the runtime low.

The Forced Coordination layout required the use of counters, so the addition of counters made evaluation of this layout in the first place possible. The layout also requires agents to fully cooperative with each other, which resulted in scores of zero for all situations where the planning was paired with an agent they could not fully predict. The achieved scores could be improved by future research by making the behavioural cloning models used in evaluation more realistic. Currently they are very prone to loop the same actions forever, especially on more complex layouts like Forced Coordination. Possible ways this can be achieved are adding some randomness into the agents or adding a memory, so they have a limited learning capacity.



(a)



(b)

Figure 7: (a) Rewards for all model-based planning agent variations for all layouts except Forced Coordination and Counter Circuit. All variations were evaluated on all layouts except for the agent with counters, which was only evaluated on the Cramped Room layout due to time constraints; and the agent with history, which was only evaluated on Cramped Room and partially on Asymmetric Advantages, which is missing the P_{BC}^{Hist} results, also due to time constraints. (b) Average states expanded per timestep for the same agents and layouts as (a). Notice the logarithmic scale, used to properly show the extreme differences between some agents. The amount of states expanded is linearly correlated with the runtime and machine independent, and was therefore used as a more objective measure of the impact on runtime.

References

[1] A. Dafoe, E. Hughes, Y. Bachrach, T. Collins, K. R. McKee, J. Z. Leibo, K. Larson, and T. Graepel, "Open problems in cooperative ai," *arXiv preprint arXiv:2012.08630*, 2020.

[2] S. M. LaValle, *Planning algorithms*. Cambridge university press, 2006.

[3] M. Carroll, R. Shah, M. K. Ho, T. Griffiths, S. Seshia, P. Abbeel, and A. Dragan, "On the utility of learning about humans for human-ai coordination," *Advances in neural information processing systems*, vol. 32, 2019.