Agent X: Improving Exploration vs Exploitation in the State of the Art Angry Birds AI

Daniel Lutalo
School of Computing
Australian National University
Canberra, Australia
Daniel.Lutalo@anu.edu.au

***Abstract*—AI agents have successfully employed deep reinforcement learning methods to surpass human performance in various new tasks over the past decade, notably including the domain of games. However, Angry Birds requires complex physical and spacial reasoning that is yet to be captured by such means. We present our logic-based Angry Birds AI which won the 2021 IJCAI AIBIRDS competition and propose a simple new method we call Second Order Thompson Sampling (SOTS) which allows for fine-tuning the balance between exploration and exploitation. We cover the competition scores of our entrant Agent X, its predecessor - the former state of the art Bambirds 2019, and the new and improved Bambirds 2021. We find that our agent has the best all-round performance, but would gain a lot by incorporating the improvements of Bambirds 2021. We list other potential areas of improvement for a future superhuman Angry Birds AI.**

***Index Terms*—Artificial Intelligence, Angry Birds, Exploration vs Exploitation**

I. Introduction

Great strides have been made to advance AI in challenging domains to a degree that far surpasses human intelligence with games like Go [6]. However, such domains are greatly restricted compared to the real physical world, and the successful types of machine learning methods cannot yet adequately capture other forms of more complex physical and spacial reasoning [4]. Angry birds serves as a useful intermediary domain that's more difficult than games like Chess and Go in such types of reasoning, yet is still not fully representative of all the challenges that will face future AI agents interacting in the physical world. Table I shows an overview of the comparison in difficulty between Go, Angry Birds, and the real world.

Go is static, i.e., the environment doesn't change while the agent is choosing an action, but the Angry Birds environment is dynamic if the agent interleaves its actions. However, Angry Birds can also mostly be treated as static if the agent waits for the level to reach a resting state before making its next move. In Go there are finite and discrete actions which an agent can make, yet in Angry Birds the action space is comprised of three continuous dimensions of choice: the launch angle, the launch force and the time to activate a bird's ability mid-flight. Furthermore, in Go the agent has full observability of its environment and can completely specify the exact state it is in, whereas in Angry Birds the state is hidden from the agent and it has to construct its own representation from an imperfect computer vision module. The biggest difference, however, between the problem of Go and Angry Birds is the model of the environment; because Go has a perfect model, it can perfectly simulate future states when considering the outcome of an action. Yet in Angry Birds, the agent can only estimate potential future states for each action arising from a complex range of interactions within the level without any access to the underlying physics engine of the game. This renders it effectively impossible to see even one move ahead and sequence multiple shots in a plan.

TABLE I: Comparison of domain difficulty

	Go	*Angry Birds*	Real World
Environment	static	semi-static	dynamic
Action space	finite	continuous^3	continuous$^{n\in\mathbb{Z}}$
Observability	full	partial	partial
Model	perfect	approximate	approximate

Fig. 1: An Angry Birds level from the 2019 Grand Final.

In Figure 1 a slight difference in any dimension of the action space (angle, force, timing of ability activation mid-flight) can be the difference between clearing the level in one shot or inadvertently piling further material to protect the remaining pig(s), and distinguishing between the two can be NP-Hard [7]. Therefore, the continuous action space with partial observability and uncertain outcomes aligns Angry Birds closer towards robotics and physical real world interaction problems compared to games like Chess and Go.

The overall goal of this competition is to build AI that surpasses human performance in Angry Birds [3], and has been considered as a next milestone in the contest between man and machine [2]. The former champion, Bambirds 2019

(no competition in 2020), entirely favours exploitation over exploration during the competition with a greedy policy which strictly adheres to its evaluation when selecting an action. However, the agent is unable to predict the exact outcome of each action, so despite having a reasonably strong evaluation engine, it's bound to contain inaccuracies which may result in a different order between the selected actions and the true best actions.

The problem we address is to provide a policy with a variable level of adherence to the evaluated scores, which we configure to lean heavily towards following the evaluation, while not being completely greedy but allowing some room for lower evaluated actions to be tried. We do this by proposing a method we call Second Order Thompson Sampling, which takes a second Thompson Sample of an exponentiated original Thompson Sample. The choice of exponent allows for a spectrum of continuous probability distributions that can represent complete exploration, complete exploitation, and infinitely many others in between. Unlike softmax which only produces superlinear probability distributions, we are able to produce sublinear, linear and superlinear distributions.

We compare Bambirds 2019[1] with our derivative agent[2] and examine the 2021 AIBIRDS competition results[3] to find that a policy including this adjustable level of adherence to the agent's evaluation of actions allows for consistent net improvement. Despite facing opposition from a much more sophisticated Bambirds 2021[4] [11], the general competency of our agent exploring different and sometimes overlooked strategies prevailed. We also discuss potential future improvements to assist the effort in surpassing the best human players.

II. Background

A. Angry Birds Game

Angry Birds is a popular 2D physics-based simulation puzzle game, with the goal to eliminate all the pigs whilst maximising points for each level. This is done by shooting birds from a sling to either strike the pigs directly or to impact the other components of the level which interact to eventually take out the pigs. There are a range of pigs with a span of sizes and health, different materials with a variety of strengths and qualities, and different types of birds with their own unique abilities and effects. Points are awarded for destroying blocks, eliminating pigs and minimising the number of allocated birds used, but all scores are tentative and only actually conferred once all pigs have been cleared. Given the continuous spectrum of actions available for each state, the search space is effectively infinite. Furthermore, the difficulty is compounded by the butterfly effect of all the possible complex interactions between the blocks on the level once a bird is fired and force is transferred.

[1]https://github.com/dwolter/BamBirds/releases/tag/BamBirds_2019
[2]https://github.com/dwolter/BamBirds/releases/tag/v21-AgentX
[3]http://aibirds.org/past-competitions/2021-competition/results.html
[4]https://github.com/dwolter/BamBirds/releases/tag/v21.5.2

Algorithm 1 High-level overview of Bambirds 2019

while remaining time > 0 **do**
  $level \leftarrow$ select level
  loop
    $s \leftarrow$ take screenshot
    $state \leftarrow$ apply computer vision to s
    if $state.won$ **or** $state.lost$ **then**
      exit loop
    end if
    $shots \leftarrow$ generate actions from $state$
    $shot \leftarrow$ select action from $shots$
    execute action $shot$
  end loop
  update internal record of $level$
end while

B. AI Birds Competition

In the AI Birds competition[5], there are eight custom levels for each phase, such as the semi finals and grand finals, which the agents can play repetitively within a total 30 minute duration. Each level offers a unique scenario and poses a new set of challenges, usually having multiple solutions of varying quality, alongside pitfalls which constrain the agents to make a more sophisticated series of shots in order to prevail. The competition oversees the way that the agents can interact with the game, only allowing them to receive screenshots of the game from the server, select levels to play and submit shots (angle, force, and timing of ability activation). The annual AI Birds competition has been held since 2012, with agents entered by over 60 teams from dozens of countries across the world. Every year the agents see overall improvement and most often there is a new champion that exceeds the maximum performance of the best agent in the previous year[6].

III. Related Works

Physical reasoning and physics-based simulations play a much bigger role than just in video-games or virtual environments, but many of the underlying techniques may be applicable within the physical world. Physics Simulation Games [14] offer a useful simplified platform for researchers to experiment and develop improved AI techniques which can then be applied in the real world. The type of understanding used to make Angry Birds' towers topple over could be used to prevent real Jenga towers from doing so [12], and the understanding of different material properties could be used to assist robots perform warehouse tasks [13].

In Angry Birds, the agent must perform a complex chain of activities encompassing computer vision, knowledge representation and reasoning, reasoning under uncertainty and planning. Various agents implementing a variety of techniques such as reinforcement learning, internal simulations and heuristics have been developed [15] [16] [17] [18] [19] [20] [21]. A

[5]http://aibirds.org
[6]http://aibirds.org/past-competitions.html

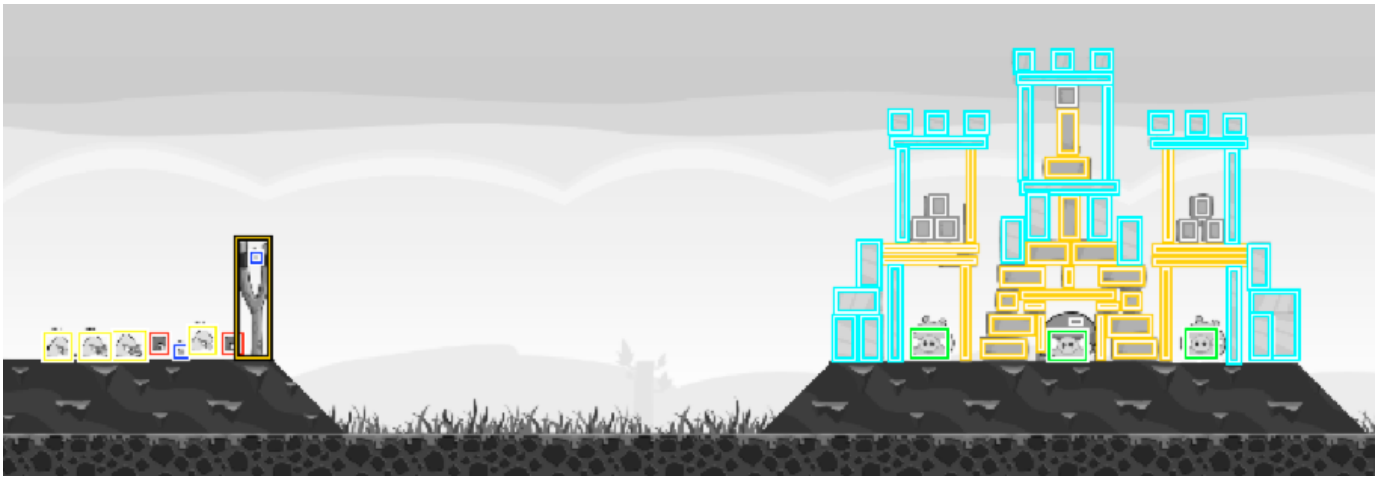


Fig. 2: An Angry Birds level with interactive features highlighted by the computer vision module.

more in-depth look at the AI Birds competition and various other agents can be found in Stephenson et al. [10] or the other agent papers, but this paper will focus on our entrant, Agent X, and the undocumented Bambirds 2019, the former state of the art, from which Agent X was derived.

A high level overview of the Bambirds algorithm is described in Algorithm 1. Although the partitioning of these agent components may be represented differently, they cover the following five areas:

- **Computer vision** to detect level features
- **Reasoning** over the relationship between level features
- **Planning** generating subgoals and actions
- **Shot Execution** enacting and evaluating actions
- **Level Selection** choosing what to try next

A. Computer Vision

The foundation of the AI Birds problem is underpinned by the computer vision module's ability to accurately reconstrue the information from each screenshot in a more compact abstract model. It takes note of the position and colour of each pixel and the similarity in relation to its neighbouring pixels in order to detect edges and boundaries. It then finds the minimum bounding rectangle to encompass shapes and analyses the content to classify the entities inside, such as ice/wood/stone block, TNT, pig or bird. It also ensures to encode the x and y coordinates of each level feature so that the spatial data is preserved and can be reasoned with. An example of this feature detection can be seen in Figure 2.

This portion of the agent remains largely unchanged from the original sample code produced by the competition hosts during its inception, and yields a high degree of accuracy for all *known* objects. However, there are new *unknown* barriers which have been recently introduced that aren't detected, and there are interesting levels involving non-convex shapes that aren't adequately prescribed within the minimum bounding rectangle. These potential areas of improvement will be discussed in the future works section.

B. Reasoning

Reasoning builds upon the knowledge contained in the abstract level model produced by the computer vision section and uses deduction and inference to generate new knowledge about how objects will behave. It analyses the material properties of the blocks and predicts the effects that gravitational and kinetic forces will have on them, also noting how blocks cover each other and limit the direct angles of approach a bird may take. The agent is able to reason as to which blocks protect the pigs, how much force they require to be removed, which blocks are on the ground, which ones are suspended upon those, how falling blocks may topple others, what can be impacted by TNT, the angles and forces that birds can strike targets, what is in reasonable range, and more.

There is a rich array of useful spatial reasoning capabilities, such as spotting round boulders, deducing that they are rollable, detecting slopes, and combining that information to infer higher level insights from simulated causal relationships. Some examples of higher level insights for Figure 2 might be that the centre pig is too well protected to eliminate with the first blue bird, the pig on the right can be taken out by the blue bird by penetrating the ice but the angle of attack is not feasible, the pig on the left has stones suspended above it and sufficiently damaging the base of the tower can cause them to fall and crush the pig, etc.

There is also a series of lower level internal physics simulation properties used to calculate the interaction of forces, such as generating possible parabola arcs and estimating the force required to break blocks or knock them over. It is by no means perfect in predicting the exact outcomes of a level after the myriad of chain reactions, yet it can still manage to produce good insights in some scenarios which even humans might not suspect, although it misses others. Figure 3 depicts the types of block-to-block force propagation.

C. Planning

Planning is very much interlinked with reasoning in AI Birds, but is mainly differentiated in this paper by decision-

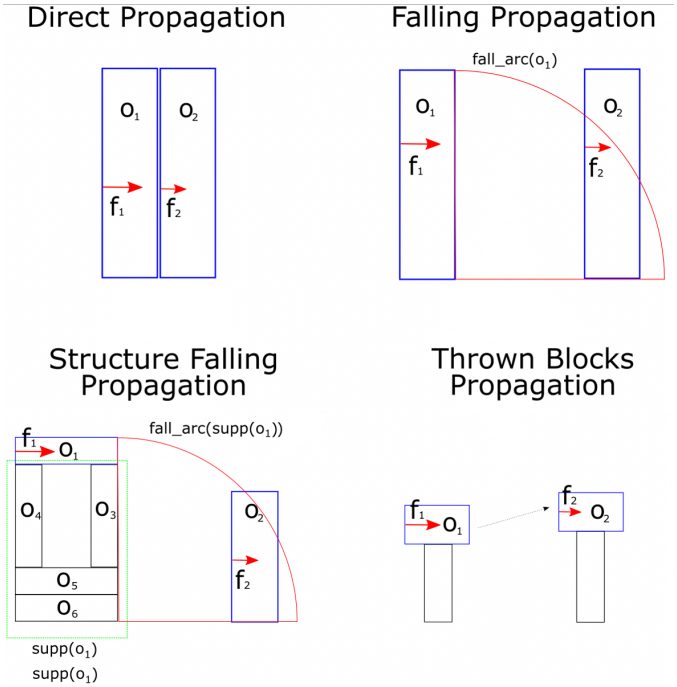


Fig. 3: Types of force propagation in Angry Birds from Liu et al. [9]

making and sequencing actions to achieve a future goal. Together they can be seen in line 9 of Algorithm 1. This is perhaps the most crucial and most difficult portion of the entire Angry Birds problem, since successive shots are rarely independent and there is a lot of uncertainty regarding the layout of a level after each shot. Based on the insights from the reasoning section, the planning component generates a collection of actions in a new layer of the planning tree and ranks them based on its evaluation, prioritising the best ones first. Due to the uncertainty in the structure of the level after each shot, the agent generates a new plan from the node in the tree it is at. If the ideas in the previous action are still feasible, the new plan should also contain them but in greater clarity, and also rank them against other unforeseen ideas that have just become discoverable to the agent.

D. Shot Execution

Shot execution covers the implementation of the actions in a plan and also the post-shot analysis of comparing expectation and reality. The result is then fed back to update the reasoning and planning cycle for dynamic learning during play. This is represented by line 11 of Algorithm 1. Currently, this only executes the selected shot, waits for the level to be settle, records the predicted vs obtained score in the node of the tree, and then cycles back to the computer vision section so the altered level can be reassessed. The main other capability of this component is the analysis of the bird's flight trajectory to possibly correct any misaligned parabolas by tuning the force and or release angle for subsequent shots. This currently only serves as the necessary glue to mediate between the agent's thoughts and actions.

E. Level Selection

The level selection component covers the higher level planning by keeping track of the remaining time and investing it in the levels believed to yield the largest improvement in total score. This is seen in lines 2 and 13 of Algorithm 1. The agent begins by first playing each level once to get a baseline reading of what's present, keeping track of the initial abstract level provided by the computer vision module. It also records the initial plans since that's almost always deterministically generated at the start, but it carries over the learned changes in its reasoning and planning on subsequent replays. It also notes the total points available if all blocks were to be destroyed, the available birds, the pigs to eliminate, the time taken to complete the first attempt of each level, and whether or not it was successful, all of which factor into future decisions on which level to play next. When each level has been played once, the level selection stochastically determines which level should be attempted again to earn the most additional points. There are pre-trained machine learning models used in predicting the expected improvement of score for each level, and a multiplier based on the estimated likelihood that the agent can actually pass the level to acquire those points.

IV. Methods

This section will cover the changes and additions made in Agent X, the new state of the art agent introduced in this paper. It will go further in-depth when comparing differences between its predecessor Bambirds 2019 as listed in the related works section. Agent X is a fork of Bambirds 2019 with changes pertaining mostly to the Shot Selection component.

A. Shot Selection: Second Order Thompson Sampling (SOTS)

Bambirds 2019 does the reasoning over each state s to generate a set of actions A along with a corresponding set of values V which represent the confidence the agent has towards the quality of the actions. Note that V doesn't necessarily represent an expected reward or expected score, but rather a level of confidence, since the potential outcome of an action is completely unknown, not merely in a stochastic sense where the transitional probabilities are unknown, but there is no knowledge of the resulting state for an unattempted action. The notation we'll use to reference the values will be the function $Q : A \to V$ such that $Q(a)$ is the value assigned to action a. Bambirds strictly adheres to the evaluation it generates by using a deterministic policy $\pi : S \to A$ that greedily selects the most promising action for each state:

$$\pi(s) = \underset{a \in A}{\operatorname{argmax}}\, Q(a)$$

Agent X introduces an alternative continuous stochastic policy π' that generalises the adherence to scores in V. Instead of using an ϵ-greedy policy [1] whereby there's a $0 < \epsilon < 1$ chance that a random move chosen and a $(1 - \epsilon)$ chance that a the greedy policy is chosen, this seeks a smoother approach which involves all values of V and not only the max.

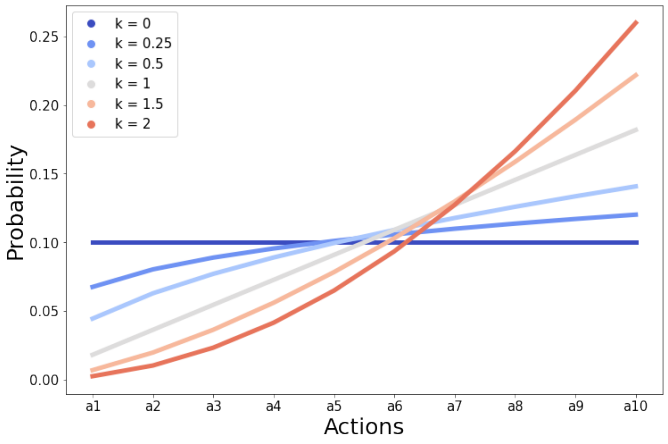


Fig. 4: SOTS probability distributions with varying exponent k

Thompson Sampling [8] applied simply in this context would give each action a a probability of being selected proportional to its value $Q(a)$.

$$P(a) = \frac{Q(a)}{\sum_{a' \in A} Q(a')}$$

However, as previously stated, the values V are not necessarily representative of expected score and we want to skew the probabilities to give greater weight according to V. Therefore, this paper introduces an additional Thompson Sample of the first Thompson Sample exponentiated, and this allows us to vary the degree of adherence toward V:

$$P'(a) = \frac{P(a)^k}{\sum_{a' \in A} P(a')^k}, \text{ where } k \in \mathbb{R} \geq 0$$

This Second Order Thompson Sampling (SOTS) method has some useful properties dependent upon k which can effectively emulate complete adherence to Q (greedy), a complete rejection of Q (uniform sample), and a continuous spectrum in between.

For an original Thompson Sample we know that $\sum_{a \in A} P(a) = 1$ and $0 \leq P(a) \leq 1 \ \forall a \in A$. Therefore, we are able to determine the following properties for our SOTS:

- $k = 1 \implies$ SOTS $\equiv$ original Thompson Sample
- $k < 1 \to 0 \implies$ SOTS $\to$ uniform distribution
- $k > 1 \to \infty \implies$ SOTS $\to$ greedy policy

To help demonstrate this, suppose we have actions $A = \{a_1, \ldots, a_{10}\}$ and corresponding $V = \{1, \ldots, 10\}$. The graph in Figure 4 displays the probabilities for our SOTS method applied to $\langle A, V \rangle$ for several values of k.

B. Action Filters

Further to this continuous SOTS method, we also propose a simple step-function method which can also emulate a fixed range of probabilities from a greedy selection to a uniform distribution. We can use a uniform distribution after filtering the actions by either selecting some constant $c \in \mathbb{R} \geq 0$

$$\{a \in A \mid \max_{a' \in A} Q(a') - Q(a) < c\}$$

or some fraction $0 \leq f \leq 1$

$$\{a \in A \mid \frac{Q(a)}{\max_{a' \in A} Q(a')} < f\}$$

- $c = 0 \ \vee f = 0 \equiv$ greedy policy
- $c \to range(V) \ \vee f \to 1 \implies$ uniform distribution

C. Remarks

These very simple yet effective methods can provide a flexible range of soft assignments which allow us to tune the balance of exploration and exploitation for our agent. Moreover, combinations of these methods to first filter the least promising actions and then applying SOTS give us even more precise control when balancing exploration and exploitation. This is useful in the AI Birds competition because there seems to be a limit to the accuracy of predicting the value of an action. As the value function is adjusted to improve some scenarios, it may become worse in others. While building higher level neural networks and other neurosymbolic AI methods haven't been adequately developed or adapted to this problem domain, seeking an equilibrium between exploration and exploitation may be optimal.

V. Results

A. 2021 Competition

TABLE II: Semi Final 2021 AI Birds Score Comparison

Level	Bambirds19	AgentX	Bambirds21
1	0	0	0
2	**45570**	**45570**	**46140**
3	**25650**	25330	**44300**
4	20870	**23640**	**24040**
5	**58620**	40670	**57900**
6	0	**33590**	**37400**
7	43440	**60510**	**60690**
8	**40890**	**40890**	**42440**
Total	235040	**270200**	**312910**

TABLE III: Grand Final 2021 AI Birds Score Comparison

Level	Bambirds19*	AgentX	Bambirds21
1	**37980**	**37980**	0
2	31100	**31150**	**32290**
3	**50100**	47170	**49420**
4	21570	**23970**	0
5	0	**31540**	0
6	22100	**29750**	**29760**
7	**37210**	35690	**36740**
8	**20080**	**20080**	**20080**
Total	220140	**257330**	168290

** Although Bambirds 2019 didn't make it to the grand final, it was run afterwards so that the results could be compared.*

Agent X was developed and tested on the eight 2019 grand final levels, the parameters set were $k = 10$ and $f = 0.8$ yielding an average result of $350,000 \pm 10,000$, whereas its predecessor Bambirds 2019 only scored 228,050 when it won the competition that year[7]. The 2021 results shown in Tables II and III show that Agent X also managed to dominate Bambirds

[7]http://aibirds.org/past-competitions/2019-competition/results.html

2019 in both the semi final[8] and the grand final[9]. In addition to the comparison of these two is Bambirds 2021, which possesses improved reasoning and demonstrated better moves not conceived by the others. The bold scores for Bambirds 2021 on the right take precedence over any bold scores for Agent X used to compare against Bambirds 2019.

B. Further Evaluations

The additional evaluations in Figure 5 and Tables IV and V were run almost a full year after the 2021 competition, and there are changes which significantly impede any direct comparison with the competition results. Without access to the older hardware originally used to test and develop this agent, these new trials were run on a 2021 14-inch Macbook Pro yielding considerable improvements across the board. In both Bambirds 2019 and Agent X, the Prolog reasoning subcomponent used to analyse each level and generate shots is now able to run much faster and achieve a lot more within the timeframe self-allocated by the agent, drastically improving the quality of suggested shots and thus the overall score.

In addition to the change in hardware, there is also a very limited number of trials in these additional evaluations, making it difficult to account for variance and draw any definitive conclusions. Nevertheless, these additional comparisons may provide some insight into the difference between the two agents. Tables IV and V show the average and maximum scores attained by both agents in 10-minute runs on each individual level to remove the impact of level selection, with an average of 15 trials per level. Figure 5 shows the total score over time for various competition stages, allowing the agent 30 minutes to participate as they please on any of the eight levels per stage. There were only five trials for each stage, the thicker middle lines reflect the mean over all trials, and the max/min lines show the most extreme aggregate values at any given time-step and are not indicative of any individual trial. Instead of using a piece-wise step function for each second, the gradient reflects the time the agent took to complete a level and plateaus indicate periods where an agent fails a level.

[8]http://aibirds.org/2021/semifinal.mp4
[9]http://aibirds.org/2021/final.mp4

VI. Discussion and Future Work

A. Competition: Agent X and Bambirds 2019

The fact that Agent X managed to consistently outperform Bambirds 2019 on unseen sets of levels provides evidence to support the improvements which can be gained by making a better trade-off between exploration and exploitation. A key point to note here is that Agent X does not make any changes to the reasoning for shot generation, but rather makes changes for selecting the order of generated shots to try first. It simply inherits the shot generation from Bambirds 2019, however takes a different approach when sifting through the search space. All things being equal, given enough time, both agents should converge to the same score. However, given the 30 minute limitation of the competition, it can be seen that Agent X makes more efficient use of the search space and yields a noticeably higher score within the time constraints.

B. Competition: Agent X and Bambirds 2021

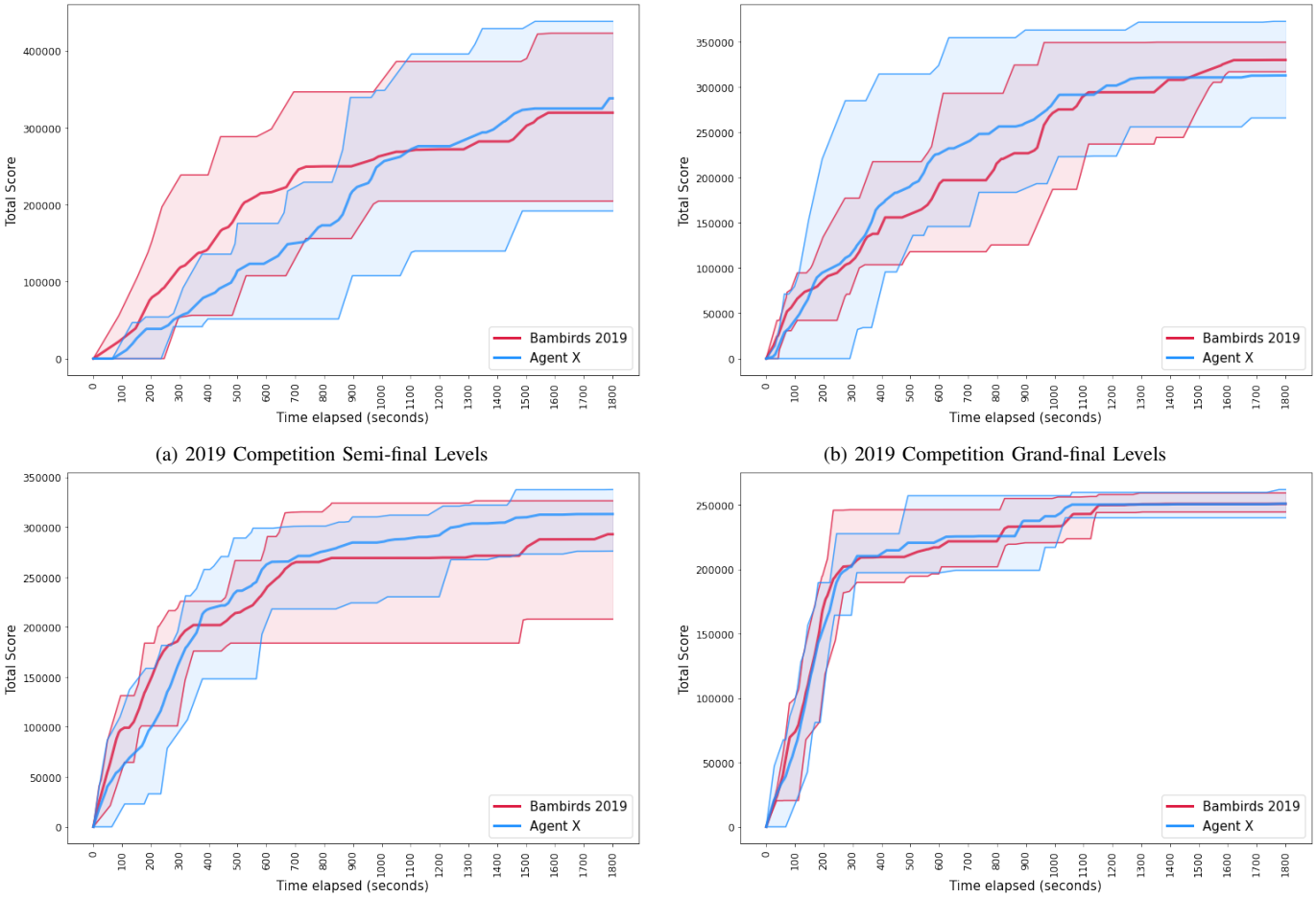
Bambirds 2021 is a significant improvement upon Bambirds 2019 and the team have developed a gamut of improved features. Bambirds 2021 enhances both the exploitation and exploration capabilities, whereas Agent X only improves exploration. The effect of this can be seen in the fact that for every level Bambirds 2021 completed, it did so with a higher score than Agent X; however, it had trouble generalising its superior insight to account for more scenarios and ultimately didn't complete as many levels as Agent X. This result may be somewhat akin to overfitting within a Machine Learning context, and highlights the difficulties in relying too heavily upon current physical reasoning capabilities. The fact that in the grand final it performed even worse than Bambirds

TABLE IV: 2019 levels - 10 minute runs

	Average Score		Max Score	
	Bam-19	AgentX	Bam-19	AgentX
SEMI-1	19236 ± 19236	**29146 ± 22074**	38600	**48370**
SEMI-2	**64212 ± 13107**	60279 ± 19073	66900	**67920**
SEMI-3	40065 ± 21449	**46469 ± 15514**	**54240**	53460
SEMI-4	**45955 ± 30084**	25038 ± 27449	**65650**	57070
SEMI-5	**48884 ± 15298**	40929 ± 19748	55570	**55610**
SEMI-6	**47102 ± 23751**	42212 ± 26565	**64780**	63990
SEMI-7	35575 ± 15909	**39484 ± 21626**	42690	**51330**
SEMI-8	0 ± 0	0 ± 0	0	0
TOTAL	**301029 ± 138837**	283559 ± 152051	388430	**397750**
FINAL-1	28620 ± 7182	**28683 ± 7178**	**32280**	30870
FINAL-2	24723 ± 24615	**31314 ± 17640**	**56910**	50530
FINAL-3	**36371 ± 10968**	23908 ± 10692	**40010**	28690
FINAL-4	**62722 ± 20044**	48352 ± 15293	**72310**	53810
FINAL-5	0 ± 0	0 ± 0	0	0
FINAL-6	**70105 ± 26142**	59962 ± 17544	**82320**	68680
FINAL-7	0 ± 0	**56535 ± 21397**	0	**65660**
FINAL-8	**30762 ± 11836**	30281 ± 18109	41210	**43030**
TOTAL	253306 ± 100790	**279038 ± 107854**	325040	**341270**

TABLE V: 2021 levels - 10 minute runs

	Average Score		Max Score	
	Bam-19	AgentX	Bam-19	AgentX
SEMI-1	**30997 ± 17896**	12728 ± 20785	41330	**46670**
SEMI-2	43400 ± 9704	**70638 ± 15252**	45570	**76110**
SEMI-3	19988 ± 10469	**44375 ± 14047**	26120	**49610**
SEMI-4	20492 ± 8982	**48042 ± 16014**	28360	**53380**
SEMI-5	**45295 ± 12567**	38586 ± 11163	**49030**	42780
SEMI-6	28088 ± 12163	**48976 ± 16925**	33500	**62530**
SEMI-7	**57879 ± 12339**	31387 ± 10186	**60510**	41230
SEMI-8	39182 ± 7836	**39240 ± 7695**	40750	40750
TOTAL	285325 ± 91960	**333975 ± 112070**	325170	**413060**
FINAL-1	36660 ± 7209	**36748 ± 7229**	38890	38890
FINAL-2	**27834 ± 7310**	27657 ± 6904	30780	**31890**
FINAL-3	44876 ± 10034	44876 ± 10034	47120	47120
FINAL-4	**9660 ± 11545**	22080 ± 5520	23460	23460
FINAL-5	18592 ± 10340	**19212 ± 10685**	25940	**25950**
FINAL-6	19535 ± 4371	**24514 ± 6823**	20720	**28110**
FINAL-7	**36366 ± 7753**	34716 ± 8058	38020	38020
FINAL-8	**19265 ± 4816**	19142 ± 4281	20470	20470
TOTAL	212792 ± 63381	**228946 ± 59536**	245400	**253910**



(a) 2019 Competition Semi-final Levels

(b) 2019 Competition Grand-final Levels

(c) 2021 Competition Semi-final Levels

(d) 2021 Competition Grand-final Levels

Fig. 5: Competition Stage Trials

2019 highlights the difficulties in improving exploitation unanimously without losing precision in other scenarios.

C. New Hardware, New Results

When run on the Apple M1 Pro, we see noticeable improvements in the scores for Bambirds 2019 in Figure 5b compared to its competition result of 228,050. In fact it performs better than Agent X on average here with a smaller variance, alluding to the intuitive idea that the original Bambirds 2019 greedy policy is preferred more as the reasoning and evaluation improves. During the development of Agent X on older hardware, it actually performed better because when it explored lower evaluated actions they would return higher scores more frequently; but now that the evaluation of actions better reflects their true score, the exploration of other actions proves worse on average - although it does provide the opportunity to occasionally select better actions.

However, when allocated more time to explore each level in Tables IV and V, Agent X still consistently performs better than Bambirds 2019, indicating that there's still value in tuning the balance between exploration and not purely relying on the complete accuracy of evaluations. Nevertheless, this is mainly speculative due to the high variance caused by the limited number of trials and exacerbated by the fact that the agents sometimes fail the level and receive a score of 0 which further skews the results.

D. Testing the New Exploration vs Exploitation Methods

These initial results from the 2021 AI Birds competition seem promising, but a more rigorous examination of the agents' non-deterministic performance on the same hardware should be done in future to account for the variance caused by randomness. Agent X's tuning parameters for SOTS and the filtering method could also be adjusted to deal with a more diverse set of levels to maximise its gains in more scenarios. These new methods could also be tested on other AI Birds agents to verify the general effectiveness, since different agent possess different strengths [5]. Given that Agent X and Bambirds 2021 both made improvements upon Bambirds 2019, there could be room to explore a fusion of the two for the best of both worlds and is something likely to be pursued in the near future. It would also be interesting to see how the exploration vs exploration methods introduced in this

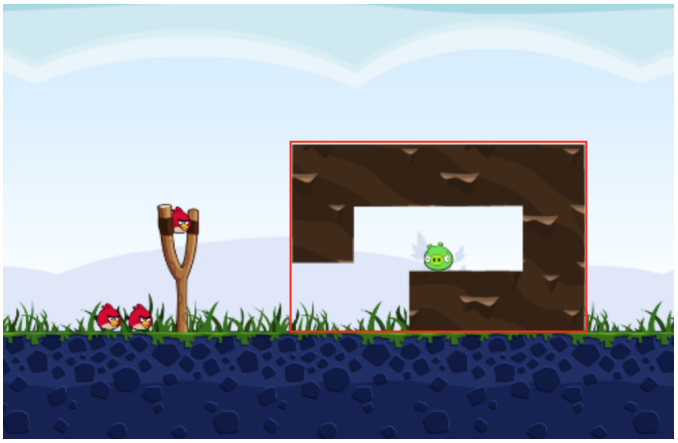


Fig. 6: An example of a difficult level from the 2019 Grand Finals beyond the current capability of Agent X

paper fare in other domains, especially for other reinforcement learning problems.

E. New AI Birds Computer Vision

An area that is due for some upgrading is the computer vision module of the agents, something nearly a decade old. There are new types of barriers which aren't identified and represented in the abstract model of the level and this partial observability creates troubles for the reasoning and planning components. This leads to odd scenarios where agents end up shooting straight into an invisible wall and only find their way around it after many attempts and much difficulty. Moreover, levels such as the one shown in Figure 6 also cannot be effectively reasoned with under the current model. If the pig is rendered as being at the centre of an indestructible material like this, the computer vision module should adapt to provide more granularity to capture these non-convex shapes, perhaps as a collection of smaller convex sub-shapes.

VII. Conclusion

This paper has covered a new simple method to provide increased fine-tune control when balancing the trade-off between exploration and exploitation. Applying it to the AI Birds problem has yielded net positive results by making more efficient use of the search space. When implemented in a new AI called Agent X, it managed to win the 2021 IJCAI competition despite facing significant opposition from competing agents with more sophisticated reasoning. Substantial advancements have been made in AI Birds since the competition inception, but further work is needed to achieve results comparable or superior to human intelligence. Development has been accelerated through collaborative open-source efforts and will likely continue as more strategies are studied and improved by competing teams.

Acknowledgements

Thanks to Jochen Renz for introducing me to AIBIRDS and the rest of the team for hosting the competition, and thanks to Diedrich Wolter and the Bambirds team for making their agent open-source.

References

[1] Barto, A. G., & Sutton, R. S. (2021). "Reinforcement Learning: An Introduction," SIAM Review Vol. 63, Issue 2 (June 2021), 63(2), 423.

[2] Grace, K., Salvatier, J., Dafoe, A., Zhang, B., & Evans, O. (2018). "When will AI exceed human performance? Evidence from AI experts," Journal of Artificial Intelligence Research, 62, 729-754.

[3] Renz, J., Ge, X., Gould, S., & Zhang, P. (2015). "The angry birds AI competition," AI Magazine, 36(2), 85-87.

[4] Renz, J., Ge, X., Stephenson, M., & Zhang, P. (2019). "AI meets angry birds," Nature Machine Intelligence, 1(7), 328-328.

[5] Shperberg, S. S., Shimony, S. E., & Yehezkel, A. (2019). "Algorithm Selection in Optimization and Application to Angry Birds," In Proceedings of the International Conference on Automated Planning and Scheduling (Vol. 29, pp. 437-445).

[6] DSilver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., ... & Hassabis, D. (2017). "Mastering the game of go without human knowledge," Nature, 550(7676), 354-359.

[7] Stephenson, M., Renz, J., & Ge, X. (2020). "The computational complexity of Angry Birds," Artificial Intelligence, 280, 103232.

[8] Thompson, W. R. (1933). "On the likelihood that one unknown probability exceeds another in view of the evidence of two samples," Biometrika, 25(3-4), 285-294.

[9] Liu, T., Renz, J., Zhang, P., & Stephenson, M. (2019). "Using Restart Heuristics to Improve Agent Performance in Angry Birds," 2019 IEEE Conference on Games (CoG) (pp. 1-8). IEEE.

[10] Stephenson, Matthew, Jochen Renz, Xiaoyu Ge, and Peng Zhang. "The 2017 AIBIRDS competition." arXiv preprint arXiv:1803.05156 (2018).

[11] Haase, F. and Wolter, D., "Behind The Corner: Using Qualitative Reasoning for Solving Angry Birds."

[12] J. Wang, P. Rogers, L. Parker, D. Brooks and M. Stilman, "Robot Jenga: Autonomous and strategic block extraction," 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2009, pp. 5248-5253.

[13] N. Correll et al., "Analysis and Observations From the First Amazon Picking Challenge," in IEEE Transactions on Automation Science and Engineering, vol. 15, no. 1, pp. 172-188, Jan. 2018.

[14] Renz, J., Ge, X. (2015). Physics Simulation Games. In: Nakatsu, R., Rauterberg, M., Ciancarini, P. (eds) Handbook of Digital Games and Entertainment Technologies. Springer, Singapore.

[15] P. A. Waga, M. Zawidzki, and T. Lechowski, "Qualitative physics in Angry Birds," IEEE Transactions on Computational Intelligence and AI in Games, vol. 8, no. 2, pp. 152–165, 2016.

[16] M. Polceanu and C. Buche, "Towards a theory-of-mind-inspired generic decision-making framework," in IJCAI Symposium on AI in Angry Birds, 2013.

[17] S. Schiffer, M. Jourenko, and G. Lakemeyer, "Akbaba: An agent for the Angry Birds AI challenge based on search and simulation," IEEE Transactions on Computational Intelligence and AI in Games, vol. 8, no. 2, pp. 116–127, 2016.

[18] F. Calimeri, M. Fink, S. Germano, A. Humenberger, G. Ianni, C. Redl, D. Stepanova, A. Tucci, and A. Wimmer, "Angry-HEX: An artificial player for Angry Birds based on declarative knowledge bases," IEEE Transactions on Computational Intelligence and AI in Games, vol. 8, no. 2, pp. 128–139, 2016.

[19] S. Dasgupta, S. Vaghela, V. Modi, and H. Kanakia, "s-Birds Avengers: A dynamic heuristic engine-based agent for the Angry Birds problem," IEEE Transactions on Computational Intelligence and AI in Games, vol. 8, no. 2, pp. 140–151, 2016.

[20] N. Tziortziotis, G. Papagiannis, and K. Blekas, "A bayesian ensemble regression framework on the Angry Birds game," IEEE Transactions on Computational Intelligence and AI in Games, vol. 8, no. 2, pp. 104–115, 2016.

[21] A. Narayan-Chen, L. Xu, and J. Shavlik, "An empirical evaluation of machine learning approaches for Angry Birds," in IJCAI Symposium on AI in Angry Birds, 2013.