Heuristic Search for Physics-Based Problems: Angry Birds in PDDL+ [Extended Abstract]*

Wiktor Piotrowski[1], Yoni Sher[1], Sachin Grover[1], Roni Stern[1,2], and Shiwali Mohan[1]
[1] Palo Alto Research Center (PARC), Palo Alto, California, USA
[2] Ben Gurion University of the Negev, Be'er Sheva, Israel
wiktorpi@parc.com, yoni.sandsher@gmail.com, sgrover@parc.com, roni.stern@gmail.com, smohan@parc.com

Introduction

Angry Birds is a very popular game that requires reasoning about sequential actions in a continuous world with discrete exogenous events. Different versions of the game are hard computationally (Stephenson, Renz, and Ge 2020), and the reigning world champion is still a human despite a long-running yearly competition in IJCAI conferences (Renz et al. 2015). In this work, we present the Hydra, the first successful game-playing agent for Angry Birds that uses a domain-independent planner and combinatorial search techniques. Hydra models the game using PDDL+ (Fox and Long 2006), a rich planning language designed for mixed discrete/continuous domains. To reason about continuous aspects of the domain, Hydra employs time discretization techniques that raise a combinatorial search challenge. To meet this challenge, we propose domain-specific heuristics and a novel "preferred states" mechanism similar to the preferred operators mechanism from classical planning (Richter and Helmert 2009). We compared Hydra with state-of-the-art Angry Birds agents (Borovicka, Spetlik, and Rymes 2014; Wang 2017). The results show Hydra can solve a greater diversity of Angry Birds levels compared to other agents and highlight its current limitations.

Background

Angry Birds consists of several different levels. Every level has some allocated numbers of birds that are launched using a slingshot. The birds are used to kill pigs sitting inside structures built using *platforms* and *blocks*, which may come in various shapes, sizes, and materials. Pigs are killed by a direct hit from a bird, or indirectly by falling blocks, explosions, or falling from a height. Every level consists of different arrangement of structures and number of pigs that have to be killed. Some birds have special abilities, activated by tapping on the screen during flight. We have not modeled them yet in Hydra for simplicity. The aim of the game is to kill the pigs using the minimum number of birds and maximize the game score. Figure 1 shows screenshots from three different levels of the game. Hydra, uses a *domain-independent* planning algorithm. Such algorithms require a *domain*, which is a model specifying the dynamics of the environment. For Angry Birds, the domain needs to model the flight of the bird, collisions between structures, explosions, and structure collapse after collisions and explosions. To define this domain, Hydra uses PDDL+ (Fox and Long 2006), an extension of the Planning Domain Description Language (PDDL) (McDermott et al. 1998) that supports numeric fluents and metrics, instantaneous actions, exogenous *events* and durative changes called *processes*.

*This paper has been accepted to ICAPS 2023.

From Angry Birds to PDDL+

Hydra obtains an Angry Birds level to play from the game's API, represented as a list of labeled objects and their locations. Then, it translates the relevant information into a PDDL+ domain and problem files. These files are passed to a domain-independent PDDL+ planner to attempt to generate a plan that kills at least one pig using one bird. If no plan is found in 30s, we execute a default non-planning action, namely a direct shot at a random pig. This continues until either the level is passed or all birds have been exhausted.

Modeling. We modeled four types of objects, namely birds, pigs, blocks, and platforms, and defined relevant state variables for the different properties of each object. For example, we model for each bird the order in which it will be fired as well as its type, location, velocity, and mass. Our PDDL+ domain model contains a single action – launching a bird from the slingshot at a chosen angle at maximum velocity. Specifically, our model includes an *increase_angle* process for increasing the launch angle and a *release_bird* action whose effects assign values to the vertical and horizontal velocity variables based on the angle. All the post-launch system dynamics are modeled using a PDDL+ process that defines the ballistics of the bird's flying, and PDDL+ events that define interactions between objects in the environment such as collisions with blocks, pigs, and platforms. For example, our model includes a ground-bounce event defining what happens when a bird hits the ground. See the full version of this paper for a complete discussion of our modeling choices. An example of the created PDDL+ problem and domain is publicly available at *https://shorturl.at/wHNQY*.

Heuristics. To guide the search, we propose two domain-specific node evaluation heuristics. The *Score heuristic*, denoted H_S, is the score returned by the game-playing agent. This score is based on how many pigs and blocks are de-

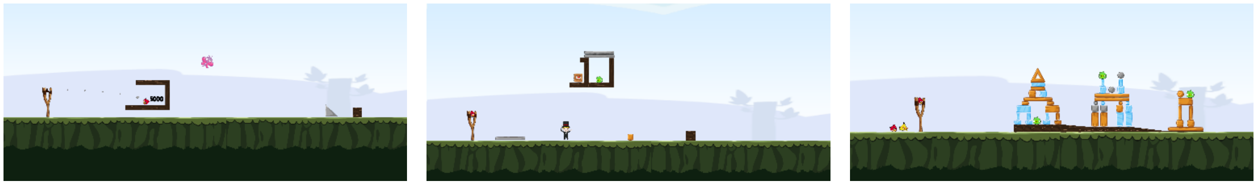

Figure 1: Screenshot of levels from type 22 (left), 57 (middle), and 55 (right). Type 22 levels can be solved by directly shooting at the pig, type 55 requires shooting a TNT block, and type 57 requires multiple shots to collapse multiple structures.

stroyed. The *Proximity heuristic*, denoted H_P, is the expected time of reaching the nearest pig given the current directed velocity under the simplifying assumption that the bird will move towards the pig in a straight line. This heuristic is admissible and gives preference to more direct shots.

Helpful States. We developed a *helpful states* mechanism for incorporating domain-specific strategies into the search process. It accepts a definition of what a helpful state is, and then prioritizes expanding such states by alternating between two open-lists, one for helpful states and one for all states. In Hydra, we defined a preferred state as one in which the active bird is on a trajectory that is expected to hit a pig or a TNT block. Checking if the active bird is on a trajectory to hit an object is done in constant time by using ballistics motion equations. To ensure the planner considers the full range of possible launch angles, we also mark all states where the bird has not yet been launched as preferred.

Experimental Results and Outlook

Our benchmark set of problems contains 8 types of *simple levels*, which contain a single bird and a relatively small number of other objects (blocks or platforms), and one *complex level* that contains multiple birds and more than 50 objects. We compared Hydra against a baseline agent designed by the IJCAI AI Birds competition organizers (ANU) (Stephenson and Renz 2017), and two former champions of the competition, namely DataLab (Borovicka, Spetlik, and Rymes 2014) and Eagle's Wing (Wang 2017). We also evaluated different versions of Hydra: breadth-first search (Hy.,BFS), depth-first search (Hy.,DFS), and greedy-best first search with the score heuristic (Hy., GBFS(H_S)), proximity heuristic (Hy., GBFS(H_P)), and helpful states (Hy., GBFS(HS)). Every agent attempted to solve 25 randomly generated levels of each type. Table 1 shows for each agent the number of levels passed from each level type in our benchmark. For Hydra ("Hyd."), we report in brackets the average number of nodes expanded, in thousands. The results show Hydra outperforms the baseline agents in all simple levels, while the baseline agents outperformed Hydra in the complex levels. In terms of expanded nodes, the helpful states are shown to be extremely effective, expanding in some cases an order fewer nodes before finding a solution (e.g., 2,760 nodes vs. 26,340 for BFS in level 53), while the heuristics are less effective.

Agent	22	25	36	45	46	53	54	57	55
Hy.,	**21**	**21**	18	**25**	18	15	11	**25**	4
BFS	(12)	(37)	(32)	(5)	(32)	(26)	(35)	(6)	(1.2)
Hy.,	**21**	**21**	20	**25**	14	15	14	**25**	1
GBFS(H_S)	(12)	(35)	(33)	(5)	(31)	(29)	(35)	(7)	(2)
Hy.,	**21**	**21**	24	**25**	**25**	**25**	**16**	**25**	4
GBFS(H_P)	(10)	(21)	(18)	(2)	(16)	(15)	(24)	(2)	(1)
Hy.,	**21**	**21**	**25**	**25**	24	**25**	**16**	**25**	9
GBFS(HS)	(8)	(10)	(2)	(2)	(3)	(3)	(16)	(2)	(1)
ANU	18	0	0	16	0	2	0	11	**17**
Datalab	15	8	0	24	3	0	0	22	16
Eaglewings	2	6	0	23	2	0	1	21	15

Table 1: Levels passed and expanded nodes (in thousands).

Discussion and Future Work. The baseline agents were developed for human-designed levels, which are solvable by a small number of fixed strategies focused on destroying complex structures. Thus, they do not perform well in domains where those strategies are ineffective, such as our simple levels which require high accuracy. Hydra performed poorly in the complex levels due to the number of objects and corresponding grounded events in them, which were as large as 138 and 13,7772, respectively. This impacts search efficiency: Hydra generated 1,610 nodes per second for type 22 levels but only 36 nodes per second for type 55 levels. Future work will focus on search strategies and heuristics that can handle such large and complex levels.

Acknowledgements

This work was partially funded by the DARPA SAIL-ON program under award number HR001120C0040

References

Borovicka, T.; Spetlik, R.; and Rymes, K. 2014. DataLab Birds Angry Birds AI. *Czech Technical University in Prague, Prague, Czech Republic, Tech. Rep. of the Datal. ab*.

Fox, M.; and Long, D. 2006. Modelling mixed discrete-continuous domains for planning. *JAIR*, 27: 235–297.

McDermott, D.; Ghallab, M.; Howe, A.; Knoblock, C.; Ram, A.; Veloso, M.; Weld, D.; and Wilkins, D. 1998. PDDL - The Planning Domain Definition Language.

Renz, J.; et al. 2015. AIBIRDS: The Angry Birds Artificial Intelligence Competition. In *AAAI*, 4326–4327.

Richter, S.; and Helmert, M. 2009. Preferred operators and deferred evaluation in satisficing planning. In *International Conference on Automated Planning and Scheduling (ICAPS)*, 273–280.

Stephenson, M.; and Renz, J. 2017. Creating a Hyper-Agent for Solving Angry Birds Levels. In *AIIDE*, 234–240.

Stephenson, M.; Renz, J.; and Ge, X. 2020. The computational complexity of Angry Birds. *Artificial Intelligence*, 280: 103232.

Wang, T. J. 2017. Description for Eagle's Wing, an AI Angry Bird Playing Agent. *University of Waterloo*.