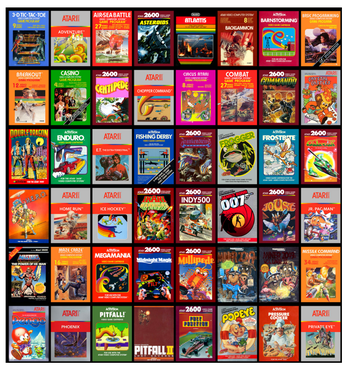


Classical Planning with Simulators: Results on the Atari Video Games

Nir Lipovetzky
The University of Melbourne
Melbourne, Australia

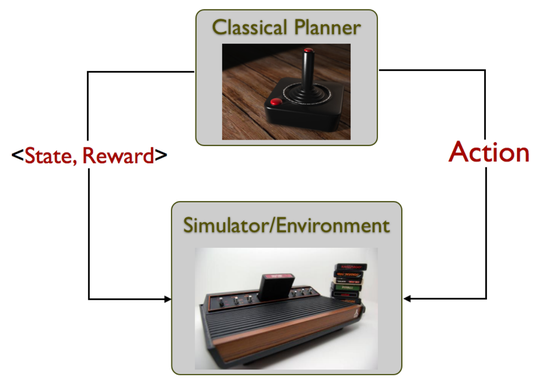
Miquel Ramirez
Australian National University
Canberra, Australia

Hector Geffner
Universitat Pompeu Fabra
Barcelona, Spain

Arcade Learning Environment (U. of Alberta)

Specs of Environment/Simulator:

- RAM: 128 Bytes
- Screen: 160 x 210 pixels with (7 bit) 128-colour palette
- Joystick: 18 actions available
- 60 frames per second. Agents choose action every 5 frames (0.08 sec) of gameplay



ALE and Classical Planning

→ Classical Planning is about achivieng goals in deterministic systems with a fully known initial state

→ Problems represented in compact, logical form with PDDL

Planning setting in ALE is deterministic and initial state fully known

Yet classical planners can't be used!

- no PDDL encoding, i.e. No action model available
- no goals but rewards

→ *Bellemare et al.* consider Breadth-first Search (*BrFS*) and *MCTS*

→ Still, some recent "classical" planning algorithms can be applied almost off-the-shelf (Lipovetzky and Geffner 2012)

A Simple Pruned Breadth-First Search Algorithm

Definition (Novelty): **smallest** new tuple of atoms generated by a state, that no other state in the search **made true before**. $novelty : s \mapsto \mathbb{N}$

- $IW(i)$ = **breadth-first** search that **prunes** newly generated states whose $novelty(s) > i$.
- IW is a **sequence of calls** $IW(i)$ for $i = 0, 1, 2, \ldots$ over problem P until problem solved or i exceeds number of vars in problem

Key theoretical property: *IW* solves P in time $O(n^i)$ if $\boldsymbol{width(P) = i}$; $IW(i)$ then solves P **optimally** (Notion of width defined by Lipovetzky and Geffner 2012)

Experiments in a nutshell: *IW*, while simple and blind, **great algorithm** when goals restricted to **single atoms**. For **multiple atomic goals**, simple serialized version *IW* suffices, achiving **atomic goals one at a time**

IW in the Atari Games

→ *IW*(1) used with the 128 state variables (RAM 8-bit word) of 256 values each

→ *IW*(1) generates then up to $128 \times 256 \times 18$ (i.e, $589,824$) states

- Children in *IW*(1) generated in random order
- Discount factor used $\gamma = 0.995$

→ Action leading to most rewarding *IW*(1)-path is executed

IW is a purely exploration algorithm that does not take into account the accumulated reward for selecting the states to consider

Balancing Exploration and Exploitation

As a simple variant that combines exploration and exploitation, we evaluated a best-first search algorithm with two queues (2BFS)

- Queue 1: ordered first by novelty measure
- Queue 2: ordered by accumulated reward

→ 2BFS alternates between the two queues (similar technique used in LAMA planner)

→ When node expanded, its children are placed on both queues, with the exception of nodes with no accumulated reward, placed only in queue 1

Experimental Results

Same setting from Bellemare et al:

- Games are played for 5 minutes maximum (18,000 frames)
- 2BFS and *IW* have a maximum lookahead budget of $150,000$ simulated frames
- UCT has same budget by running 500 rollouts of depth 300
- Score is averaged among 5 runs per game

	IW(1)		2BFS		*BrFS*	UCT
Game	Score	Time	Score	Time	Score	Score
Alien	**25634**	81	12252	81	784	7785
Amidar	**1377**	28	1090	37	5	180
Assault	953	18	827	25	414	**1512**
Asterix	153400	24	77200	27	2136	**290700**
Asteroids	**51338**	66	22168	65	3127	4661
Atlantis	159420	13	154180	71	30460	**193858**
Bank Heist	**717**	39	362	64	22	498
Battle Zone	11600	86	**330800**	87	6313	70333
Beam Rider	9108	23	**9298**	29	694	6625
Berzerk	**2096**	58	802	73	195	554
Bowling	**69**	10	50	60	26	25
Boxing	**100**	15	**100**	22	**100**	**100**
Breakout	384	4	**772**	39	1	364
Carnival	**6372**	16	5516	53	950	5132
Centipede	99207	39	94236	67	**125123**	110422
Chopper Command	10980	76	27220	73	1827	**34019**
Crazy Climber	36160	4	36940	58	37110	**98172**
Demon Attack	20116	33	16025	41	443	**28159**
Double Dunk	-14	41	21	41	-19	**24**
Elevator Action	13480	26	10820	27	730	**18100**
Enduro	**500**	66	359	38	1	286
Fishing Derby	30	39	6	62	-92	**38**
Freeway	**31**	32	23	61	0	0
Frostbite	902	12	**2672**	38	137	271
Gopher	18256	19	15808	53	1019	**20560**
Gravitar	3920	62	**5980**	62	395	2850

Game	*IW*(1) Score	*IW*(1) Time	2BFS Score	2BFS Time	*BrFS* Score	UCT Score
Hero	**12985**	37	11524	69	1324	12860
Ice Hockey	**55**	89	49	89	-9	39
James Bond	**23070**	0	10080	30	25	330
Journey Escape	40080	38	**40600**	67	1327	7683
Kangaroo	**8760**	8	5320	31	90	1990
Krull	**6030**	28	4884	42	3089	5037
Kung Fu Master	**63780**	21	42180	43	12127	48855
Montezuma Revenge	0	14	**540**	39	0	0
Ms Pacman	21695	21	18927	23	1709	**22336**
Name This Game	9354	14	8304	25	5699	**15410**
Pong	**21**	17	**21**	35	-21	**21**
Pooyan	11225	8	10760	16	910	**17763**
Private Eye	-99	18	**2544**	44	58	100
Q*Bert	3705	11	11680	35	133	**17343**
Riverraid	**5694**	18	5062	37	2179	4449
Road Runner	**94940**	25	68500	41	245	38725
Robot Tank	**68**	34	52	34	2	50
Seaquest	**14272**	25	6138	33	288	5132
Space Invaders	2877	21	**3974**	34	112	2718
Star Gunner	1540	19	**4660**	18	1345	1207
Tennis	**24**	21	**24**	36	-24	3
Time Pilot	35000	9	36180	29	4064	**63855**
Tutankham	172	15	204	34	64	**226**
Up And Down	**110036**	12	54820	14	746	74474
Venture	**1200**	22	980	35	0	0
Video Pinball	**388712**	43	62075	43	55567	254748
Wizard Of Wor	**121060**	25	81500	27	3309	105500
Zaxxon	**29240**	34	15680	31	0	22610

	IW(1)	2BFS	*BrFS*	UCT
# Times Best (54 games)	26	13	1	19
# Times Better than *IW*	–	16	1	19
# Times Better than 2BFS	34	–	1	25
# Times Better than UCT	31	26	1	–

Search Tree Depths

- *BrFS* search tree results in a lookahead of 0.3 seconds
- *IW*(1) and 2BFS result in lookahead of up to 6–22 seconds

IW Planning vs. Google DeepMind learning (Nature 2015)

- lookahead/RAM state vs. reactive controller/Screen state
- Score-wise IW better than DeepMind in 45 out of 49 games

Wrap Up

→ *IW* makes use of the state structure (atoms) to order exploration

→ Exploitation of this structure pays off in classical planning and ALE

- Similar results reported for General Video Game Playing (GVG-AI) where games are stochastic (MDPS; IJCAI GIGA-2015 Workshop)

Key benefit of *IW* w.r.t. classical planning algorithms: Does not need compact PDDL-like encodings; it can work with simulators!