Planning with Pixels in (Almost) Real Time

Wilmer Bandres[1] Blai Bonet[2] Hector Geffner[3]

[1]Universitat Pompeu Fabra, Barcelona, Spain
[2]Universidad Simón Bolívar, Caracas, Venezuela
[3]ICREA & Universitat Pompeu Fabra, Barcelona, Spain

Planning in the Atari Games

Arcade Learning Environment (ALE) (Bellemare et al., 2013) is platform that supports learning and planning settings

In planning, next move selected after **looking ahead with simulator**

Monte-Carlo Tree Search best planner initially, then IW(1):

- IW(1) uses **RAM** and takes tens of secs/decision **(not real time)**
- **New! Rollout IW(1)** that plays Atari from **screen pixels**

Standard Evaluation Setting

- 60 frames per second, **frameskip of 15**
- Episodes of at most 5 minutes (18,000 frames)
- Random number of no-ops (between 1 and 30) applied at start
- **Deterministic simulator** (`repeat-action-probability` is 0)
- Budget of half second for Rollout IW(1) **(almost real time)**
- Each data point is average of 5 runs

(Partial) Results on 49 Games

				RAS Rollout IW(1)
Game	Human	DQN	Sarsa-Blob-PROST	budget 0.5s
asterix	8,503.0	6,012.0	3,996.6	**48,700.0**
asteroids	**13,157.0**	1,629.0	1,759.5	4,486.0
bowling	**154.8**	42.4	65.9	51.6
boxing	4.3	71.8	**89.4**	78.6
breakout	31.8	**401.2**	52.9	79.8
centipede	11,963.0	8,309.0	3,903.3	**46,661.4**
gravitar	**2,672.0**	306.7	1,231.8	2,410.0
hero	**25,673.0**	19,950.0	13,690.3	11,480.0
....				
montezuma's revenge	**4,367.0**	0.0	778.1	100.0
space invaders	1,652.0	**1,976.0**	844.8	1,812.0
star gunner	10,250.0	**57,997.0**	1,227.7	15,960.0
up n down	9,082.0	8,456.0	19,533.0	**36,936.0**
venture	**1,188.0**	380.0	244.5	80.0
video pinball	17,298.0	42,684.0	9,783.9	**188,604.4**
zaxxon	9,173.0	4,977.0	8,204.4	**18,700.0**
# ≥ Human	n/a	23 (46.9%)	18 (36.7%)	**25 (51.0%)**
# ≥ 75% Human	n/a	27 (55.1%)	22 (44.8%)	**29 (59.1%)**
# best in game	**16 (32.6%)**	12 (24.4%)	6 (12.2%)	15 (30.6%)

Bold = Best   Red = Better than human   **Bold Red** = Best/better than human

Online Planning with Iterated Width (IW)

IW(1) is a breadth-first search (BrFS) where nodes that don't make a **boolean feature true for the first time** in the search are **pruned**

- Number of expanded nodes is **linear in number of features**
- Set F of **boolean features** is given
  - Classical planning: features are propositional atoms
  - Previous work in Atari: features obtained from 128 bytes of RAM
- IW(k) like IW(1) but with F replaced by conjunctions of up to k features; number of expanded nodes is $O(|F|^k)$

Pixel Features (Liang et. al, AAMAS-2016)

ALE's sensory input is 160×210 pixels (pixels of 128 colors)

- Screen split into 16×14 **disjoint tiles**, each one is 10×15 pixel patch
- **$\sim$28k Basic features:** tell which colors contain each tile
- **$\sim$6.8m B-PROS:** relative distance between tiles with 2 given colors
- **$\sim$13.7m B-PROT:** relative distance between tiles with 2 given colors at current and previous screens
- **$\sim$20.5m B-PROST:** Basic + B-PROS + B-PROT (no blob features!)

B-PROT and B-PROST contain **non-Markovian** features

Rollout IW(1)

Rollout IW(1) performs **rollouts** instead of breadth-first search

Starting in tree with only root node r (for current state), a sequence of rollouts from r is "thrown" to define lookahead tree

Pruning of nodes takes into consideration:

– Features made true in node

– Depth of node

– Minimum depth so far where each feature has been seen

Nodes are also **labeled as SOLVED**: algorithm **terminates** when root node is SOLVED or time's up

Properties of Rollout IW(1)

Theorem

- *Length of rollouts is bounded by $|F|$*
- *Each rollout improves depth to some f, or labels a node as SOLVED*
- *Root is SOLVED in at most $b \times |F|^2$ rollouts (b is branching factor)*

If IW(1) is run until completion, it is more efficient than Rollout IW(1) for reaching all **features of width 1** (see paper for def.)

Value of Rollout IW(1) is **anytime behaviour** (i.e. operation under time bound): BFS exploration replaced by rollouts that **"dive in tree"**

Extensions and Variations

- **Caching:** previous look-ahead tree partially used for next decision
- **Penalties for deaths (Risk Aversion):** death signal translated into high penalty
- **Subscoring:** “novelty” measured relative to $\lfloor \log(\text{acc. score node}) \rfloor$ instead of globally

In experiments, **RAS Rollout IW(1)** is Rollout IW(1) with these variations

Wrap Up and Future Work

- New algorithm **Rollout IW(1)** that “emulates” IW(1) in poly time, but **better anytime properties**
- Rollout IW(1) plays Atari in almost real time from screen pixels with **performance comparable** to Human, DQN, and Sarsa-Blob
- **Code:** `https://github.com/bonetblai/rollout-iw`

Future work:

- Rollout IW(k) for **noisy Atari (MDPs)**
- Use of IW(k) planners inside **Approx Modified PI** a la AlphaZero instead of MCTS

Discussion: Rollout IW(1) vs. Rainbow (54 Games)

	RAS Rollout IW(1)		
Game	budget 0.5s	budget 32s	Rainbow
alien	8,550.0	**19,354.0**	9,491.7
amidar	1,161.0	1,609.0	5,131.2
assault	264.6	281.4	14,198.5
asterix	48,700.0	87,600.0	428,200.3
asteroids	**4,486.0**	**7,344.0**	2,712.8
atlantis	113,460.0	134,660.0	826,659.5
bank heist	268.0	**2,179.0**	1,358.0
battle zone	56,200.0	**509,400.0**	62,010.0
beam rider	3,729.2	4,921.2	16,850.2
berzerk	966.0	1,640.0	2,545.6
bowling	**51.6**	**48.0**	30.0
boxing	78.6	80.2	99.6
breakout	79.8	370.0	417.5
centipede	**46,661.4**	**84,226.0**	8,167.3
chopper command	8,900.0	**33,220.0**	16,654.0
crazy climber	38,120.0	42,720.0	168,788.5
defender	**298,010.0**	**387,010.0**	55,105.0

Discussion: Rollout IW(1) vs. Rainbow (54 Games)

	RAS Rollout IW(1)		
Game	budget 0.5s	budget 32s	Rainbow
demon attack	5,201.0	9,898.0	111,185.2
double dunk	-4.0	**16.0**	-0.3
enduro	137.4	330.8	2,152.9
fishing derby	-61.8	-53.0	31.3
freeway	3.6	10.0	34.0
frostbite	1,494.0	5,970.0	9,590.5
gopher	7,256.0	11,840.0	70,354.6
gravitar	**2,410.0**	**5,540.0**	1,419.3
hero	11,480.0	29,708.0	55,887.4
ice hockey	**5.2**	**18.2**	1.1
kangaroo	1,800.0	5,280.0	14,637.5
krull	1,645.2	2,837.0	8,741.5
kung fu master	2,980.0	24,300.0	52,181.0
montezuma revenge	100.0	1,080.0	384.0
ms pacman	**13,746.8**	**21,833.0**	5,380.4
name this game	6,128.0	6,820.0	13,136.0
phoenix	5,386.0	7,570.0	108,528.6

Discussion: Rollout IW(1) vs. Rainbow (54 Games)

	RAS Rollout IW(1)		
Game	budget 0.5s	budget 32s	Rainbow
pitfall	-814.6	-692.4	0.0
pong	-1.4	18.2	20.9
private eye	2,160.0	-340.0	4,234.0
qbert	14,160.0	**40,350.0**	33,817.5
road runner	25,780.0	**62,960.0**	62,041.0
robotank	30.0	51.8	61.4
seaquest	1,236.0	9,846.0	15,898.9
skiing	-15,806.6	-15,473.6	-12,957.8
solaris	1,620.0	1,728.0	3,560.3
space invaders	1,812.0	4,362.0	18,789.0
star gunner	15,960.0	17,160.0	127,029.0
surround	– na –	– na –	9.7
tennis	**3.2**	-3.4	-0.0
time pilot	8,540.0	5,480.0	12,926.0
tutankham	147.4	191.2	241.0
venture	**80.0**	**120.0**	5.5
video pinball	188,604.4	375,073.0	533,936.5
wizard of wor	**40,780.0**	**75,380.0**	17,862.5
yars revenge	3,647.8	10,523.6	102,557.0
zaxxon	18,700.0	**38,700.0**	22,209.5
# better than Rainbow	**10 (18.5%)**	**17 (31.4%)**	