Plannig 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 is selected after **looking ahead with simulator**

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

- Planning using **RAM states** in tens of secs/decision **(not real time)**
- **New! Rollout IW(1)** that plays Atari from **screen pixels**
- **Simulator:** 60 frames per sec, frameskip of 15, **budget of 0.5 sec** per decision **(almost real time)**

(Partial) Results on 49 Games

Game	Human	DQN	Sarsa-Blob-PROST	RAS Rollout IW(1) 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
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
james bond	406.7	576.7	636.3	**5,340.0**
.....				
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**
# $\geq$ Human	n/a	23 (46.9%)	18 (36.7%)	**25 (51.0%)**
# $\geq$ 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 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$13.7m B-PROT:** relative distance between tiles with 2 given colors at current and previous time points
- **$\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* $|F|^2 \times b$ *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**

Value of Rollout IW(1) is **anytime behaviour** (i.e. operation under time bound) as breadth exploration is 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" in IW(1) relative to $\lfloor \log(\text{acc. score node}) \rfloor$; in classical planning relative to **number of achieved goals**

Rollout IW(1) with last 2 variations denoted by **RAS Rollout IW(1)**

(Partial) Results on 49 Games

Game	Human	DQN	Sarsa-Blob-PROST	RAS Rollout IW(1) 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
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
james bond	406.7	576.7	636.3	**5,340.0**
.....				
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**
# $\geq$ Human	n/a	23 (46.9%)	18 (36.7%)	**25 (51.0%)**
# $\geq$ 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

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** with Human, DQN, and Sarsa-Blob

Future work:

- Rollout IW(k) for **noisy Atari (MDPs)**
- Use of IW(k) planners inside **Approx Modified PI** a la AlphaZero instead of MCTS
- Analysis of **visual features** in relation to **width**