

RHEA Enhancements in General Video Game Playing

IEEE CIG 2017

General video game playing

- One agent plays many games
- Games may be unknown / previously unseen

General Video Game AI (GVGAI, gvgai.net)

- Real-time planning problem (40ms)
- 2D grid-based games
- No information about game rules
- Limited information about **current** game state only
- **Forward Model** for simulation of possible future states



Motivation

- Rolling Horizon Evolutionary Algorithms (RHEA) show promise
 - in general video game playing
 - as showcased in GVGAI
- Several improvements in literature in various contexts
 - do they work in GVGAI?
 - do they work **together**?

Methodology

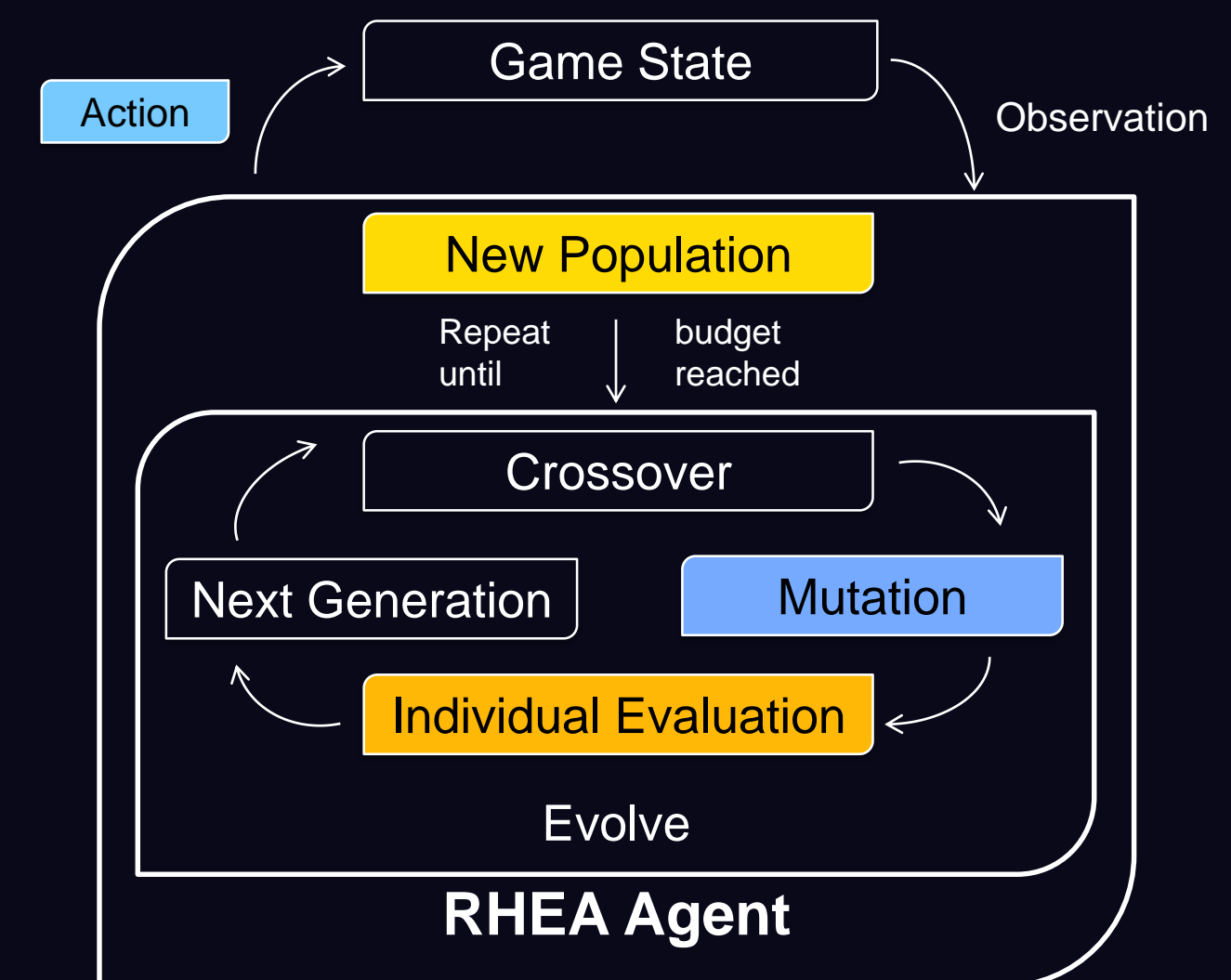
- Look at 4 parts in the evolution process
 - Mutation operator
 - Population management
 - Action recommendation policy
 - Individual evaluation
- In **isolation** and **combined** (hybrids)
- Split into 2-part experiment
- On 20 GVGAI games
- With different core parameter configurations

Experimental Setup

- Population size P - Individual length L = {1-6, 2-8, 5-10, 10-14}
- Budget: 900 Forward Model calls
- First part:
 - EA-bandit, EA-tree and EA-shift (plus hybrids)
- Second part (best from first part + rollouts):
 - EA-tree, EA-shift and EA-roll (plus hybrids)
- Validation
 - Comparison with MCTS

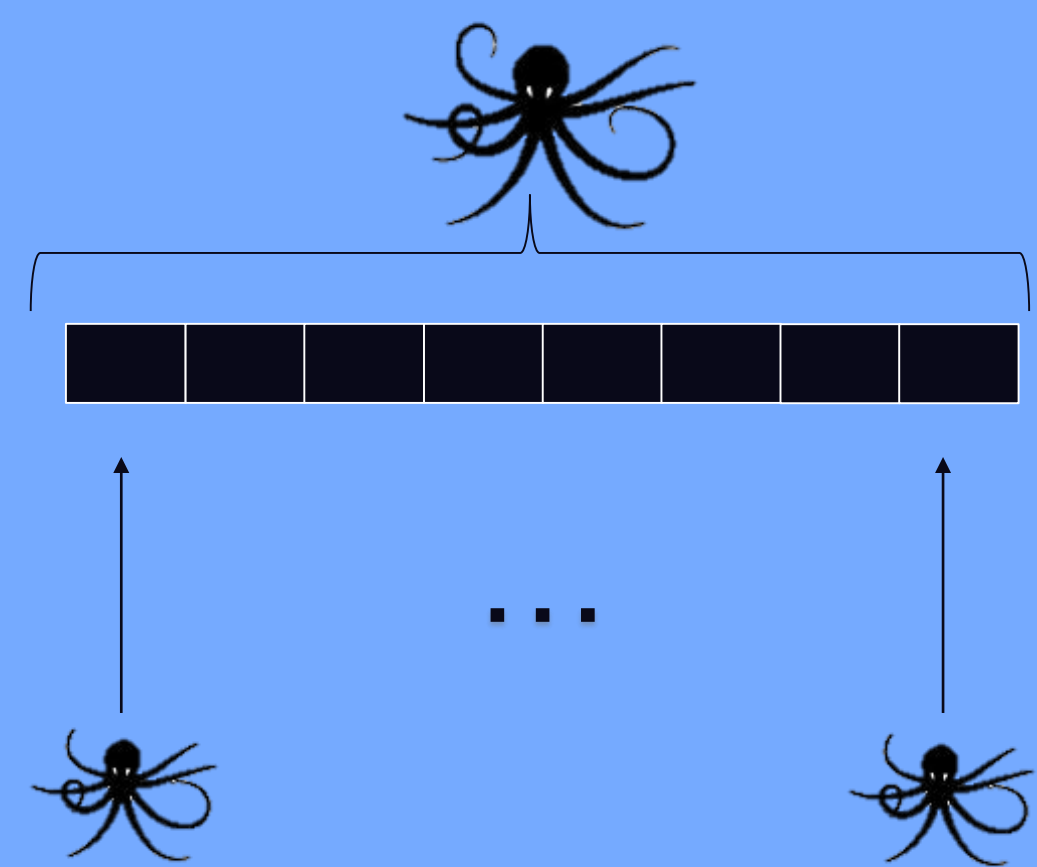
Results Overview

- Shift buffer best, Bandit mutation worst
- Performance proportional to parameter values
 - But algorithm ranking not stable
- Shift buffer (small params) better than best vanilla



EA-Bandit

- Bandit-based mutation operator
- Balance between exploration and exploitation
- 2 level UCB bandits
 - **Individual level:** which gene?
 - **Gene level:** which value?
- $UCB1 = \underset{a \in EA(s)}{\operatorname{argmax}} \left\{ Q(s, a) + C \sqrt{\frac{\ln N(s)}{N(s, a)}} \right\}$

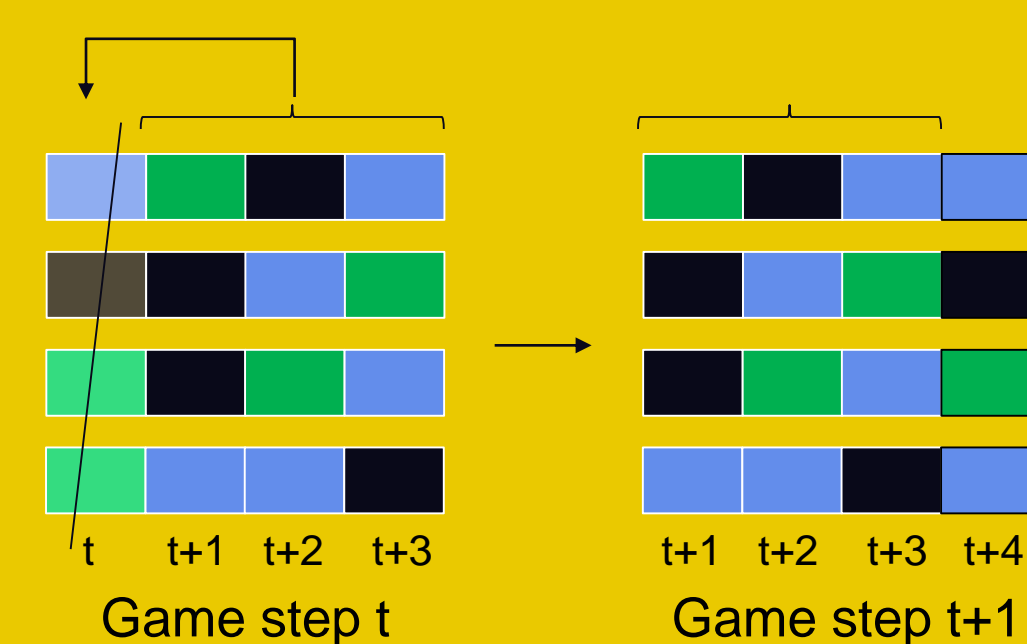


Results

- One of worst variants
- Epistasis (does not work for sequences of actions in this form)
- Most beneficial in large configurations

EA-Shift

- Shift buffer for population management
- Keep population between game ticks
- Shift population to the left at next tick
- Add new random action at the end



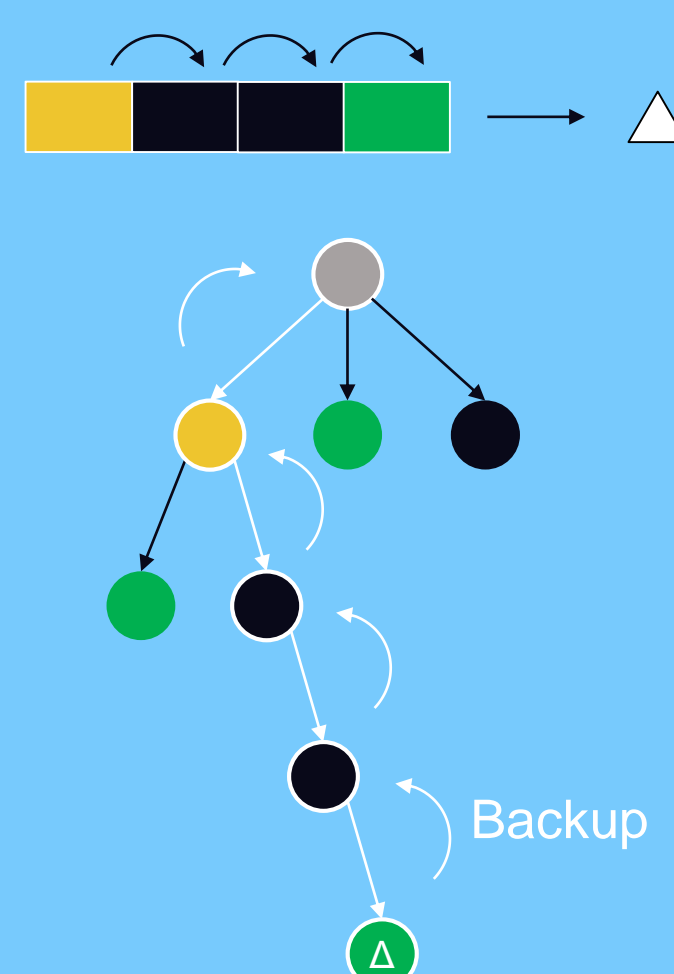
Results

- Best variant
- Higher scores in most games
- 1-6: tree hybrids better
- Worst hybrids: +bandit mutation

#	Algorithm	F1 Points	Avg. Wins
1	EA-Shift-Roll	430	42.05 (2.48)
2	MCTS	430	41.30 (1.76)

EA-Tree

- During evaluation, keep **statistics** in a tree structure
- Similar to Monte Carlo Tree Search
 - Tree only used to **choose action**
- Final action: most visited (top level)

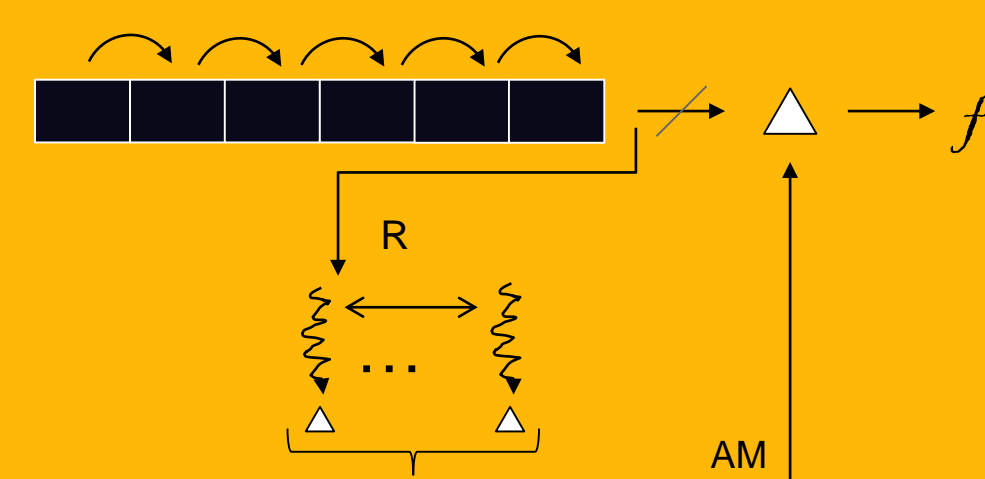


Results

- Mid-table
- Worst hybrids: +bandit mutation
- Most beneficial in low configurations

EA-Roll

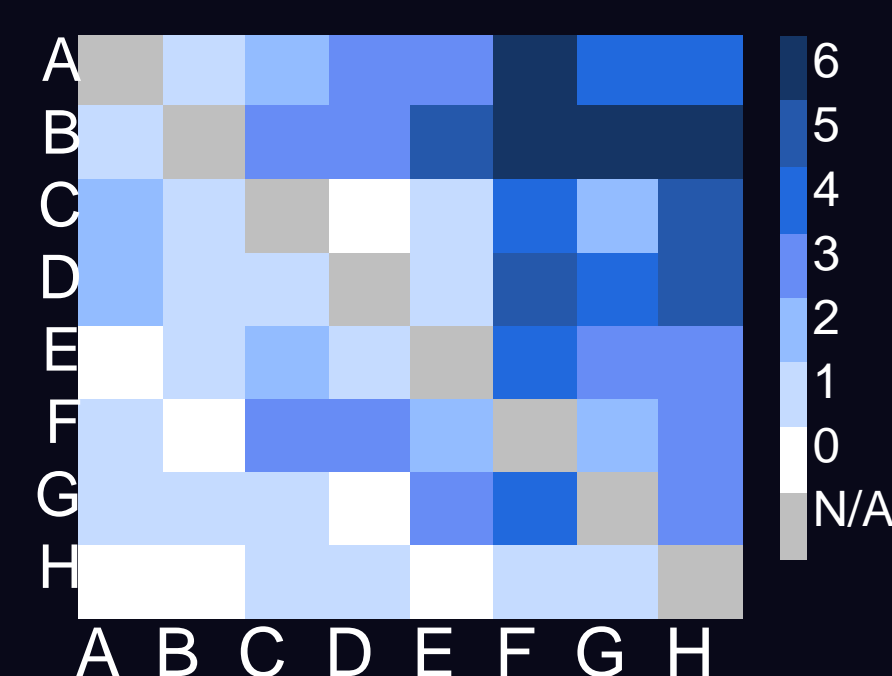
- At the end of individual evaluation
 - Monte Carlo simulation
 - Length L/2.
- Repeat R times
 - Use avg. as individual fitness



Results

- Best: EA-Shift-Roll (10-14, R=5), matches MCTS
- Most beneficial in low configs
- Most variants better with rollouts
- Tree hybrids worse

#	Algorithm	F1 Points	Avg. Wins
1	MCTS	451	41.30 (1.76)
2	EA-Tree-Roll	409	35.90 (2.27)

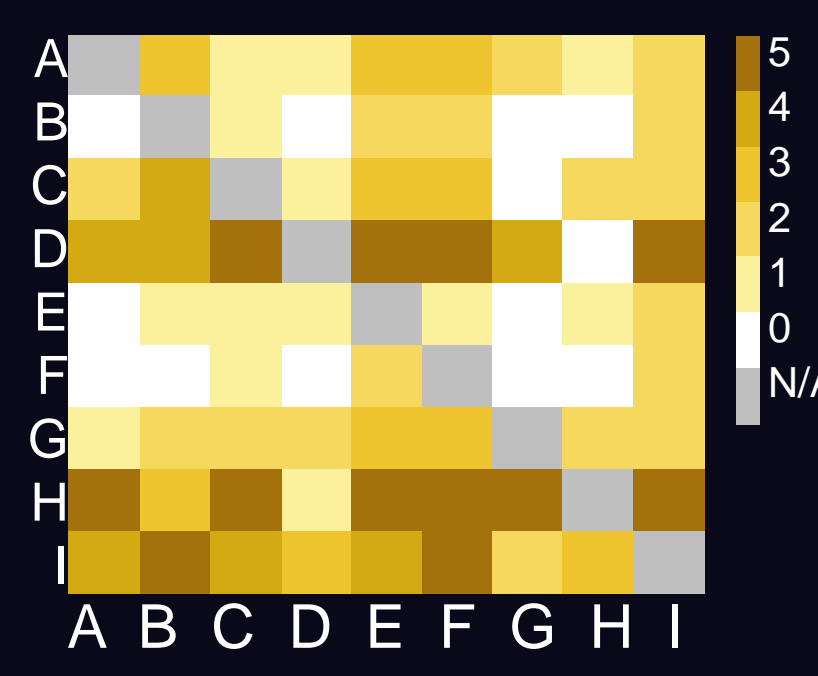


Win percentage for config 5-10. Color bar: in how many unique games row was significantly better than column.

A Vanilla E EA-Bandit
B EA-Shift F EA-Bandit-Shift
C EA-Tree G EA-Bandit-Tree
D EA-Tree-Shift H EA-Bandit-Tree-Shift

Future work

- Meta-heuristic: which one best for this task?
- Improved bandit mutation
- More games to better judge significance



Win percentage for config 10-14, R=5. Color bar: in how many unique games row was significantly better than column.

A Vanilla F EA-Tree-Roll
B EA-Roll G EA-Tree-Shift
C EA-Shift H EA-Tree-Shift-Roll
D EA-Shift-Roll I MCTS
E EA-Tree