Analysis of Rolling Horizon Evolution Parameters in General Video Game Playing

Raluca D. Gaina, Jialin Liu, Simon M. Lucas, Diego Perez-Liebana

Introduction

- One of the most promising techniques in General Video Game AI competition (GVGAI) are the Rolling Horizon Evolutionary Algorithms (RHEA).
- Analysis of the vanilla version of RHEA on 20 GVGAI games
  - Special focus on the population size and the individual length.
- Comparison with the sample Monte Carlo Tree Search (MCTS)
  - Best sample agent in GVGAI.
  - Base of many winning competition entries.

RHEA in Game AI Literature

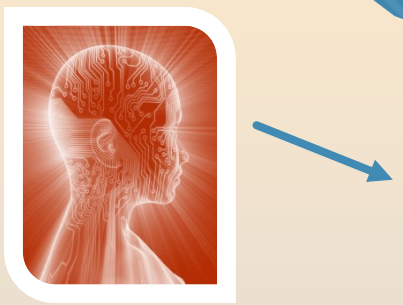
- Perez et al: comparison with tree search on the Physical Travelling Salesman Problem
- Justesen et al: Hero Academy, groups of actions evolved for a single turn, for up to 6 different units, fixed population of 100 individuals (online evolution is able to beat MCTS).
- Wang et al: modified version in Starcraft micro[1], evolving plans to determine which script each unit should use at each time step.

Hero Academy: https://youtu.be/nox2dk0_aSA


Starcraft micro: https://youtu.be/Xpjp0sm2reE

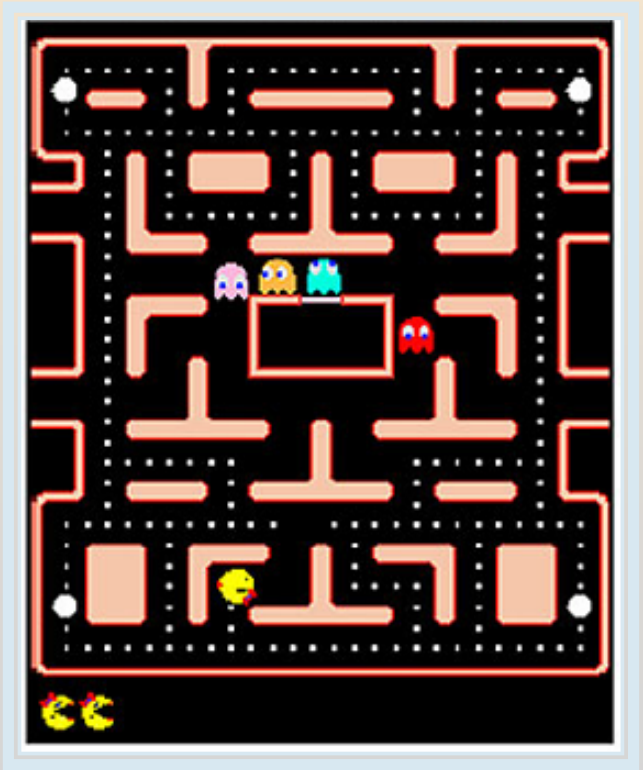
[1] JarCraft: https://github.com/tbalint/JarCraft ::: SparCraft: https://github.com/davechurchill/ualbertabot

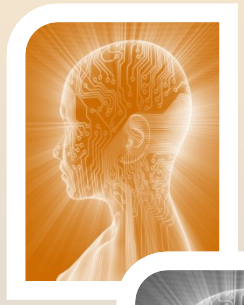
Game AI

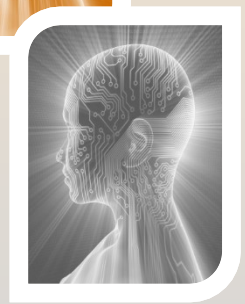


Super Mario AI

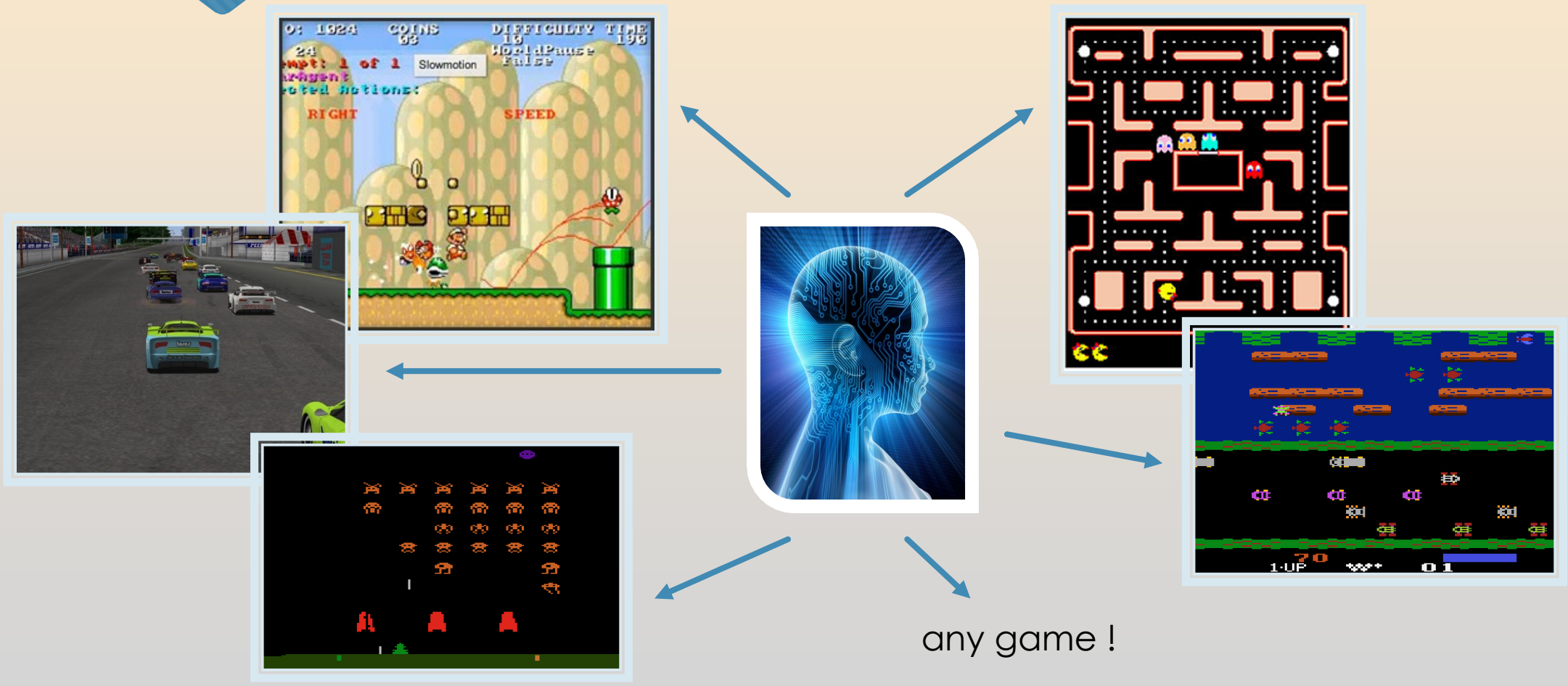
Ms. Pacman







General Video Game AI



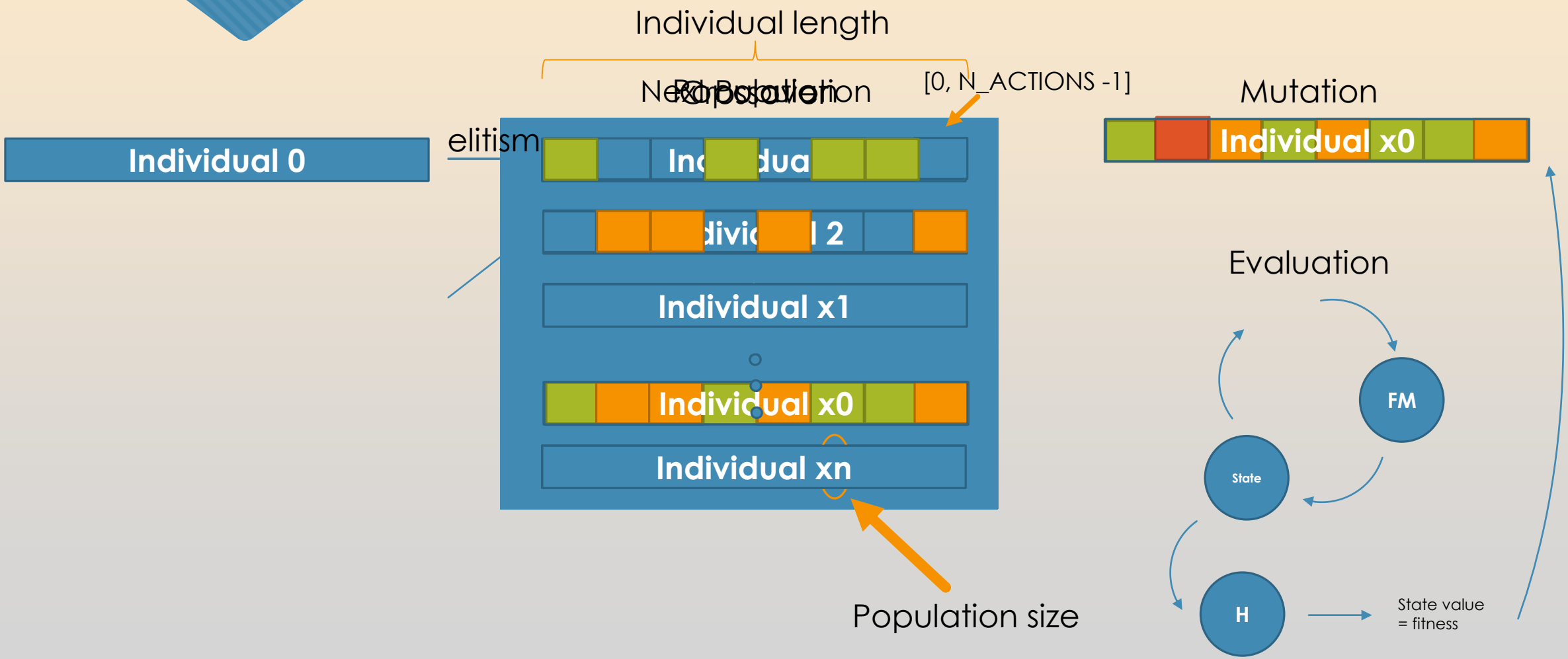
any game !

General Video Game AI Competition

- 2D grid-physics games
- Arcade, puzzles, shooters, adventure.
  - Ways to interact with the environment
  - Ways to win
  - Elements in a game
  - Scoring systems
  - Single and two player, cooperative and competitive.

... agents receive only a high-level view of the current game state and must make decisions in real-time (40ms)

Rolling Horizon Evolution



Approach

- Population sizes P={1, 2, 5, 7, 10, 13, 20}, individual lengths L={6, 8, 10, 12, 14, 16, 20}
- All other parameters fixed to default values
- Budget: 480 Forward Model calls
- Special case tested – Random Search: P=24, L=20
  - No evolution.
- Validation
  - Comparison with MCTS.
  - Budget extension.

20 Games from GVGAI corpus

Survive Zombies

rmly sampled from two
roller performance in 62
rollers + competition ent

nced set: 10 stochastic,

MCTS

Aliens

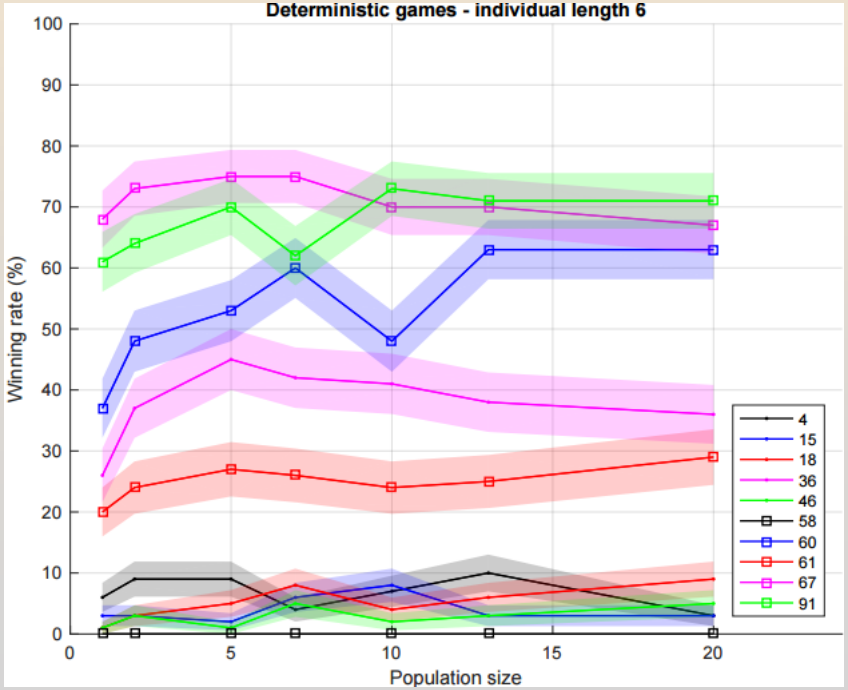
Sea Quest

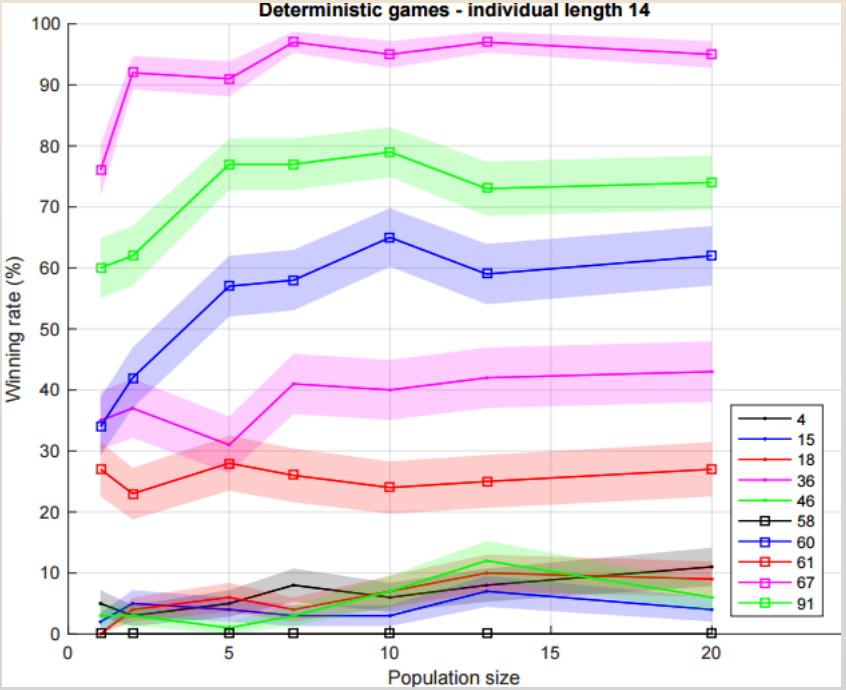
Results Overview

- Trend noticed in most of the games: win rate increases, regardless of game type. Overall, performance increases with greater parameter values.
- Exceptions:
  - win rate starts at 100% (room for improvement, *Aliens* and *Intersection*)
  - win rate stays very close to 0% (outstanding difficulty, *Roguelike*).
- Best: **P** = **20**, **L** = **20**
  - 47.50 (2.33) win rate
- Worst: **P** = **1**, **L** = **20**
  - 33.15 (2.60)

Results – Population Variation (Deterministic)

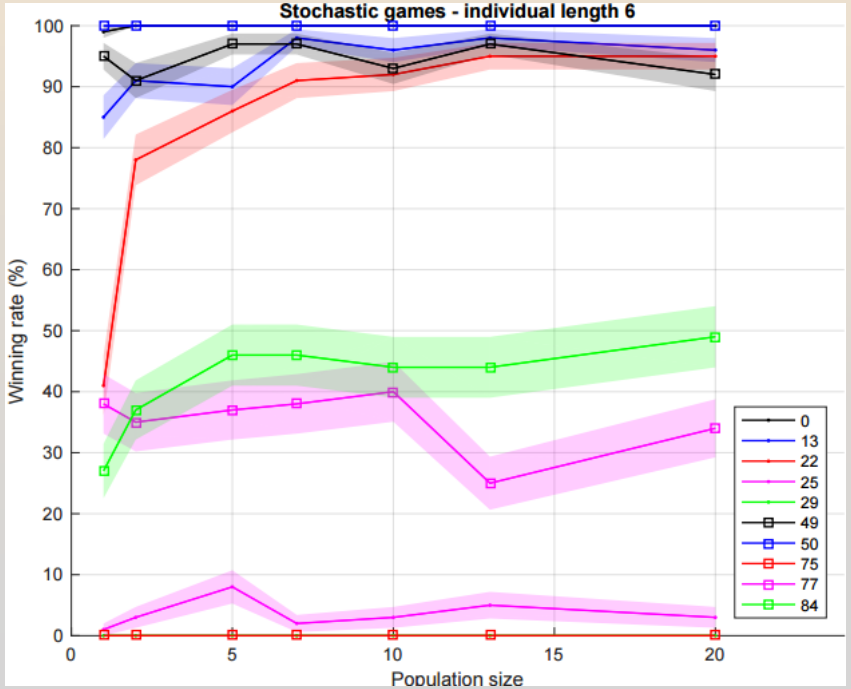
- Winning rate increases progressively in most games.
- High diversity in performance
- Interesting games (largest performance difference):
  - Game 67 (Plaque Attack)
  - Game 91 (Wait for Breakfast)
  - Game 60 (Missile Command)

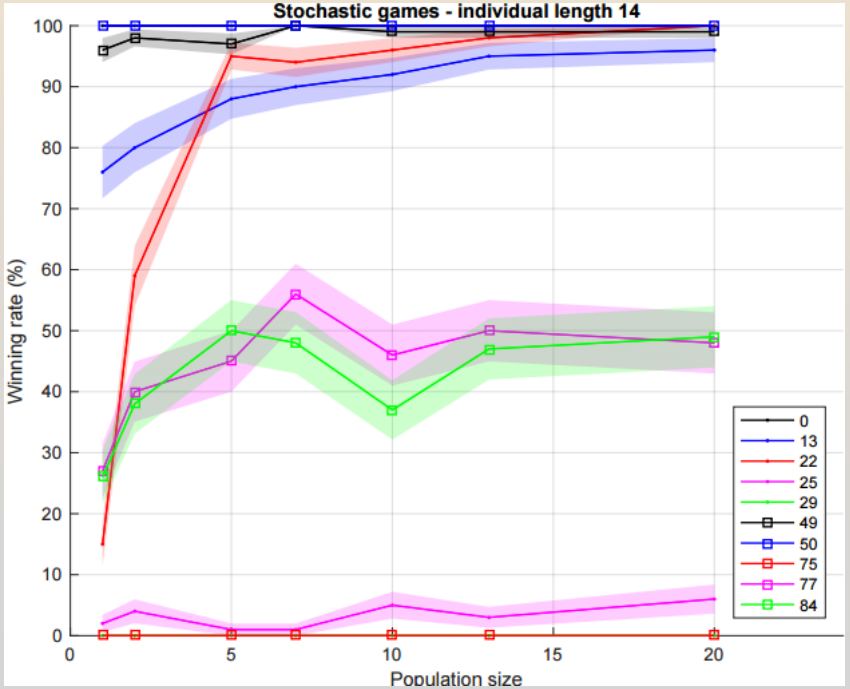




Results – Population Variation (Stochastic)

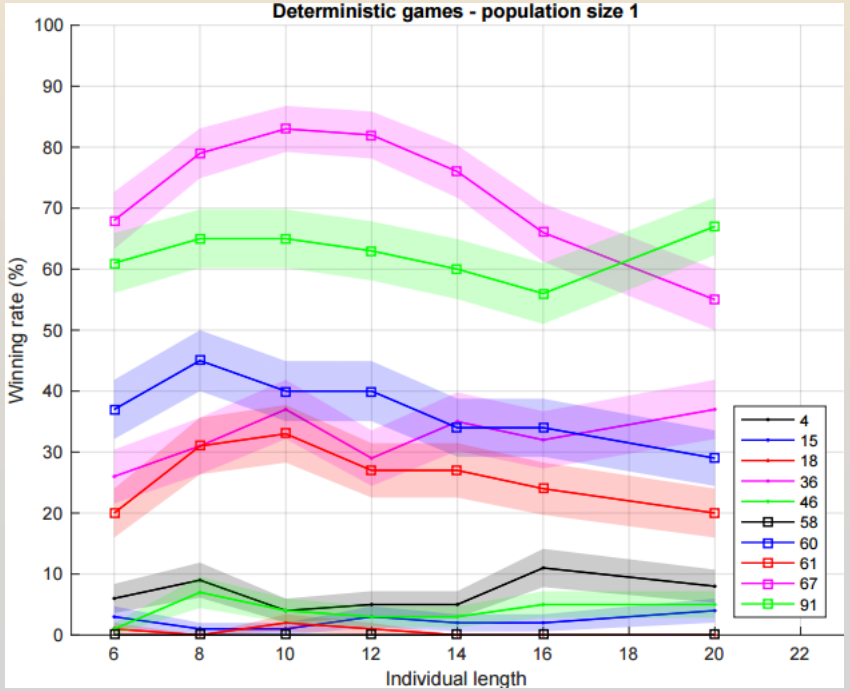
- If the length of the individual is small, increasing the population size is not beneficial in all cases, sometimes causing a drop in win rate
- Interesting games (largest performance difference):
  - Game 13 (Butterflies)
  - Game 22 (Chopper)
  - Game 25 (Crossfire)
  - Game 77 (Sea Quest)
  - Game 84 (Survive Zombies)

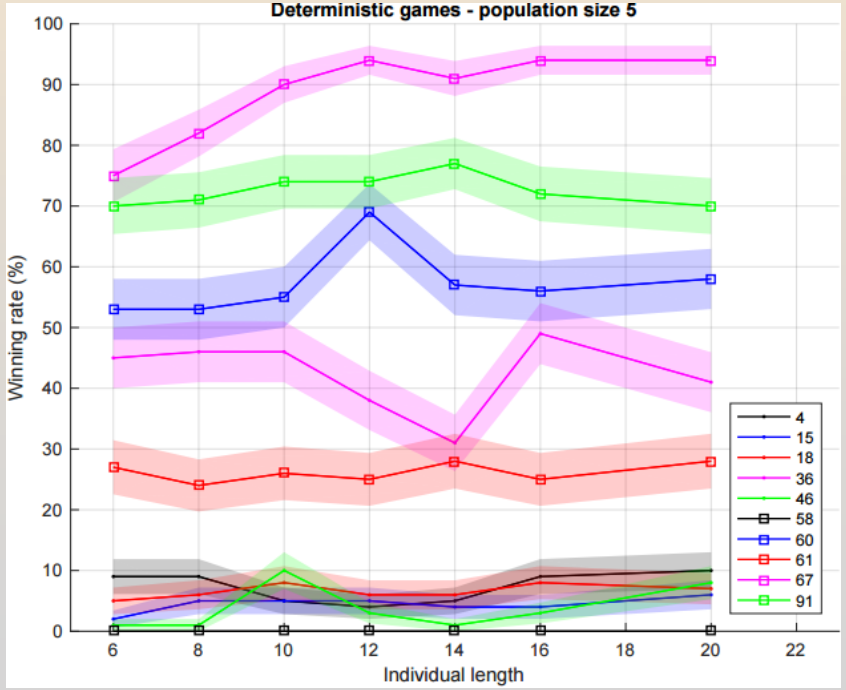




Results – Individual Variation (Deterministic)

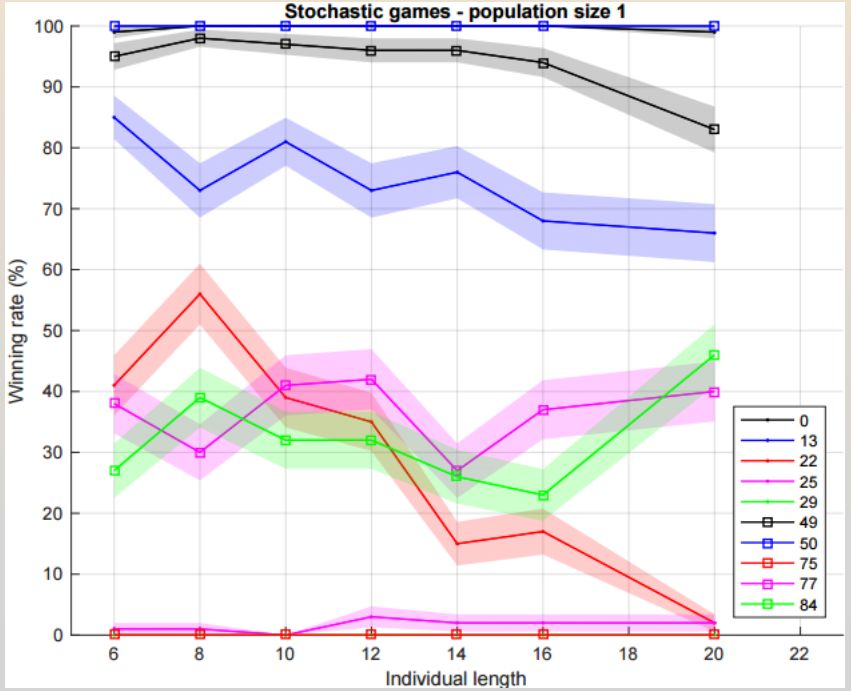
- If population size is small, win rate sees a significant increase followed by a drop in large individual lengths; this issue is solved by increasing the population size.
- Interesting games (largest performance difference):
  - Game 67 (Plaque Attack)

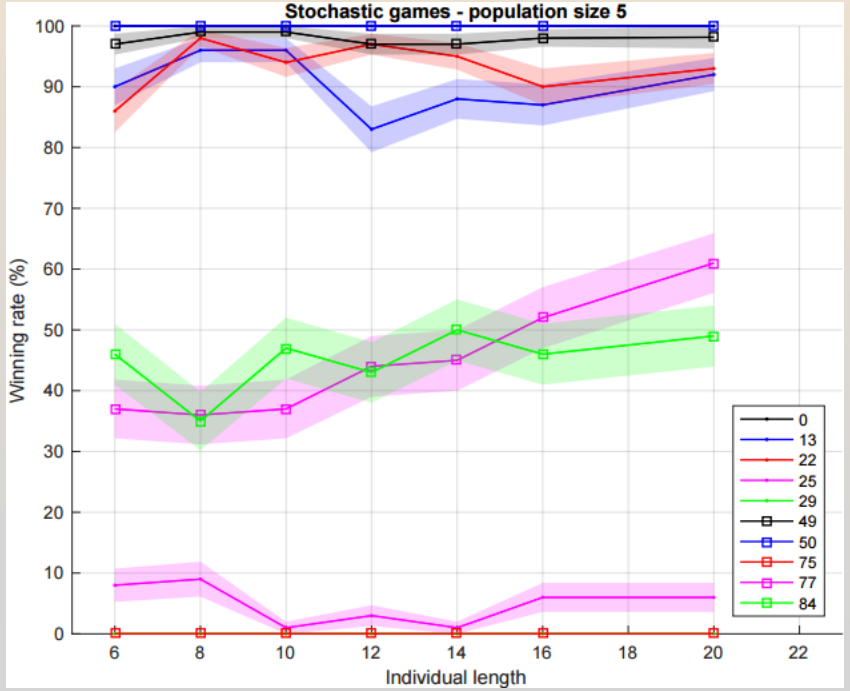




Results – Individual Variation (Stochastic)

- Performance highly dependant on game.
- No significant change in win rate can be appreciated in larger population sizes.
- Interesting games (largest performance difference):
  - Game 13 (Butterflies)
  - Game 22 (Chopper)





Results – Random Search & Increased Budget

- Reminder:
  - No evolution.
  - L = 20, P = 24
- Performance no worse than any other RHEA configuration.
- Budget increase => Performance increase

Algorithm	Average Wins (T)	Points (T)	Average Wins (D)	Points (D)	Average Wins (S)	Points (S)
RHEA-1920	48.25 (2.36)	351	36.30 (2.88)	181	60.20 (1.84)	170
RHEA-1440	48.05 (2.23)	339	35.40 (2.82)	177	60.70 (1.65)	162
RHEA-960	47.85 (2.39)	323	34.60 (2.99)	162	61.10 (1.79)	161
RHEA/RS-480	**46.60 (2.40)**	**271**	**32.90 (3.04)**	**131**	**60.30 (1.76)**	**140**

Results – RHEA vs MCTS

- If P > 5, RHEA outperforms MCTS.
- Random Search (RS) outperforms MCTS in terms of win rate, but not in F1 points.
  - MCTS is more general.
- In deterministic games, MCTS performance similar to *worst RHEA* configuration (P=1, L=20).
- In stochastic games, MCTS and RS performances are similar.

Algorithm	Average Wins (T)	Average Wins (D)	Average Wins (S)
Worst RHEA	*33.15 (2.60)*	*22.50 (2.99)*	*43.80 (2.22)*
RHEA P=1	37.95 (2.47)	26.90 (2.93)	49.00 (2.01)
RHEA P=2	41.05 (2.62)	27.90 (3.05)	54.20 (2.20)
RHEA P=5	44.65 (2.40)	31.90 (3.18)	57.40 (1.61)
RHEA P=7	44.65 (2.36)	30.80 (3.09)	58.50 (1.64)
RHEA P=10	44.06 (2.26)	29.50 (2.90)	58.60 (1.63)
RHEA P=13	45.15 (2.47)	32.10 (3.06)	58.20 (1.88)
RHEA P=20	44.75 (2.31)	31.50 (2.87)	58.00 (1.74)
RS	46.60 (2.40)	32.90 (3.04)	60.30 (1.76)
MCTS	**41.45 (1.89)**	**22.20 (2.45)**	**60.70 (1.34)**

Summary

- Analysis of population size and individual length of vanilla Rolling Horizon Evolutionary Algorithm (RHEA)
- Win rate measured on 20 games of the General Video Game AI corpus (selected based on difficulty for a diverse set, deterministic vs stochastic).
- Special case of Random Search studied, comparison with MCTS and increased budget effects.

Conclusions

- RHEA is no better than Random Search, worse in many cases.
- RHEA cannot explore space quickly enough in limited budget (the increased budget results confirm this; so better and faster evolutionary operators and improvements are needed).
- RHEA can outperform MCTS if population size is high.
- Performance increased in most games in higher population sizes and higher individual lengths, but there are cases where the opposite is true.
- Bigger impact noticed in population size variation than individual length.

Future Work

- Meta-heuristics: devise methods to identify the type of game being played and …
  - … employ different parameter settings.
  - … modify dynamically parameter settings.
- Improvement of vanilla RHEA in this general setting.
  - Seeking bigger improvements of action sequences during the evolution phase, without the need of having too broad an exploration as in the case of RS.
  - Being able to better handle long individual lengths in order for them to not hinder the evolutionary process.
- Consider effects in stochastic games of …
  - … More elite members.
  - … Resampling individuals to reduce noise.

Thank you