# General Video Game playing AI: A Survey

Wangzhihui Mei 2019124044

CCNU-UOW JI

*maywzh@gmail.com*

December 25, 2019

# Overview


- Introduction
  - Research Scope
  - Motivation
- Landscapes
  - Knowledge Representation
  - Search
  - Planning
  - Learning
  - Generality
- Approachs
  - Neural Network
  - Reinforcement Learning
  - Monte-Carlo Tree Search
- Conclusion

# Video Game Playing AI

Video games have long been popular benchmarks for Artificial Intelligence(AI)[1]. Many researches have been done in building a optimal **Video Game Playing AI(VPAI)** in certain games. Such attempts comprises Chess, Go, Car Racing games, Ms.PacMan, Real-Time Strategy (RTS) games and Super Mario Bros, etc.

# Research Scope

In this research, we focus on **General Video Game Playing AI(GVPAI)**.
GVPAI aims at building an no-human-intervening game-playing agent that is able to playing multiple games rather than specialized VPAI designed for certain one game[1].

# Motivation[2]

The objective of General Video Game Playing (GVGP) is to by-pass the addition of game specific knowledge, especially if the algorithm is tested in games that have not been played before.
Obviously, algorithms that approach GVGP problems may still count on some kind of domain knowledge, and the questions raised above could still be asked. Indeed, many different algorithms can be employed for GVGP, and chances are that heuristics will still make a big difference.
However, by reducing the game-dependent knowledge, approaches are forced to be more general, and research conducted in this field is closer to the open domain of General Artificial Intelligence.

# Landscapes of GVPAI

- Knowledge Representation
- Search
- Planning
- Learning
- Generality

# Knowledge Representation and Reasoning[3]

GVPAI requires a formal, symbolic language in which the rules of arbitrary games can be described to a system. The general Game Description Language (GDL) has been developed for that purpose. There are several challenging reasoning problems.

- Reasoning about actions
- Automated theorem proving
- Other KRR techniques

# Search[12]

Once a GVPAI system is capable of computing legal moves and position updates from the game rules, it can search through the space of pos- sible ways in which the game can proceed

- Monte Carlo Tree Search
- Informed search – Uses problem-specific knowledge.

# Planning

Planning is closely related to GVGP, as both are instances of general problem solving, where the specifics of a problem are unknown until runtime.

# Learning

The very idea of general game playing is to build systems that automatically learn to master arbitrary new games.

# Generality[4]

**Heading**

1. Game generality
2. Task generality
3. Player generality

Develop AI methods that work with not just one game, but with multiple game.
Develop methods that can do not only one task but several differnet related tasks.
Develop methods that can model, respond or reproduce the large variability among humans in design style, playing style, preferences and abilities.

# Approachs

- Neural Network(NN)
- Reinforcement Learning(RL)
- Monte-Carlo Tree Search(MCTS)

# NN-based GVGP

M. Hausknecht et al. [5] employed evolutionary neural networks to extract higher-dimensional representation forms from the raw game screen.
All algorithms encode Artificial Neural Networks (ANNs) which are represented by weights and connectivity(topology).

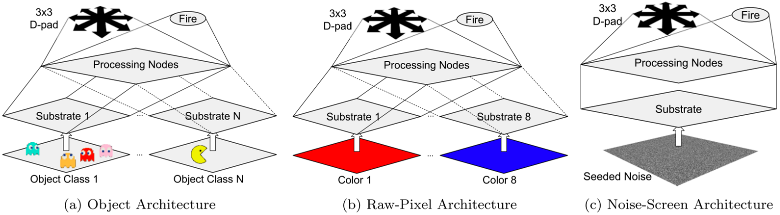


Figure: ANN topology for 3 different state representation

# RL-based GVGP I

Yavar Naddaf [6] used 2 main approaches: RL-based and search-based methods. RL-based methods use feature vectors generated from the game screen as well as the console RAM to learn to play a given game. The search-based methods use the emulator to simulate the consequence of actions into the future, aiming to play as well as possible by only exploring a very small fraction of the state-space.

Figure: Conceptual figure of feature vector generation for the game Freeway

# RL-based GVGP II

Bellemare et al. [7] explored the concept of contingency awareness (the recognition that a future observation is under an agent's control and not solely determined by the environment) using Atari 2600 games.
The research introduced a technique for accurately identifying contingent regions and describe how to exploit this knowledge to generate improved features for value function approximation.

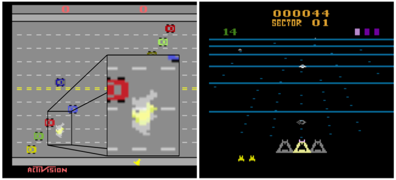


Figure: The contingent regions, shown by a transparent overlay, for Freeway (left) and Beam Rider (right)

# MCTS-based GVGP I

Perez et. al. [8] explored the performance of a vanilla Monte Carlo Tree Search algorithm, and analyzed the main difficulties encountered when tackling this kind of scenarios and did some modifications to overcome these issues, strengthening the algorithm's ability to gather and discover knowledge, and taking advantage of past experiences.

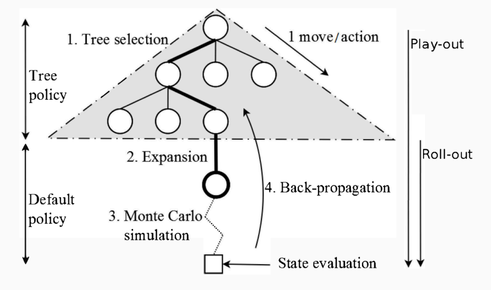


Figure: MCTS algorithm steps

# MCTS-based GVGP II

Park and Kim[9] propose to use influence map (IM) to solve the MCTS's horizon effect problem.

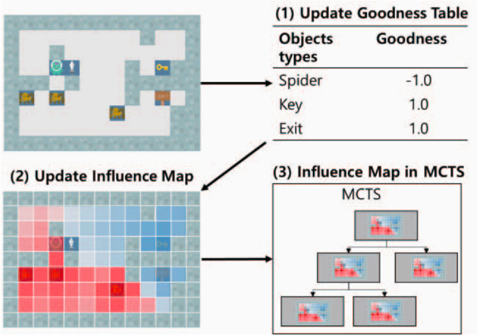


Figure: Overview of proposed method (blue is good influence and red is bad influence)

# MCTS-based GVGP III

Sironi et.al.[10] designed three Self-Adaptive MCTS (SA-MCTS) agents that optimize the parameters of a standard non-Self-Adaptive MCTS agent of GVPAI.
The three agents select the parameter values using Naive Monte-Carlo, an Evolutionary Algorithm and an N-Tuple Bandit Evolutionary Algorithm respectively, and are tested on 20 single-player games.

# MCTS-based GVGP IV

| Games | sampleMCTS | SA-MCTS$_{NMC}$ | SA-MCTS$_{EA}$ | SA-MCTS$_{NEA}$ |
|---|---|---|---|---|
| Aliens | **100.0(±0.00)** | 99.8(±0.39) | **100.0(±0.00)** | 99.4(±0.68) |
| Bait | 6.6(±2.18) | 7.0(±2.24) | 7.8(±2.35) | **8.4(±2.43)** |
| Butterflies | 95.2(±1.88) | 95.0(±1.91) | 94.2(±2.05) | **95.4(±1.84)** |
| Camel Race | 4.2(±1.76) | 4.6(±1.84) | **6.2(±2.12)** | 5.2(±1.95) |
| Chase | 3.2(±1.54) | 7.2(±2.27) | **9.2(±2.54)** | 7.4(±2.30) |
| Chopper | **91.4(±2.46)** | 88.6(±2.79) | 83.2(±3.28) | 50.8(±4.39) |
| Crossfire | 4.2(±1.76) | 11.6(±2.81) | 11.4(±2.79) | **15.6(±3.18)** |
| Dig Dug | 0.0(±0.00) | 0.0(±0.00) | 0.0(±0.00) | 0.0(±0.00) |
| Escape | 0.2(±0.39) | 4.4(±1.80) | 7.6(±2.33) | **13.0(±2.95)** |
| Hungry Birds | **5.4(±1.98)** | 2.6(±1.40) | 4.6(±1.84) | 3.8(±1.68) |
| Infection | 97.0(±1.50) | 95.6(±1.80) | 97.6(±1.34) | **97.8(±1.29)** |
| Intersection | 100.0(±0.00) | 100.0(±0.00) | 100.0(±0.00) | 100.0(±0.00) |
| Lemmings | 0.0(±0.00) | 0.0(±0.00) | 0.0(±0.00) | 0.0(±0.00) |
| Missile Command | 60.4(±4.29) | **60.8(±4.28)** | 58.0(±4.33) | 58.6(±4.32) |
| Modality | 27.0(±3.90) | 27.4(±3.91) | 26.0(±3.85) | **28.4(±3.96)** |
| Plaque Attack | 91.8(±2.41) | 92.0(±2.38) | **92.8(±2.27)** | 92.6(±2.30) |
| Roguelike | 0.0(±0.00) | 0.0(±0.00) | 0.0(±0.00) | 0.0(±0.00) |
| Sea Quest | 55.0(±4.37) | 47.8(±4.38) | **55.6(±4.36)** | 43.2(±4.35) |
| Survive Zombies | **41.0(±4.32)** | **41.0(±4.32)** | 34.8(±4.18) | 34.8(±4.18) |
| Wait for Breakfast | 15.4(±3.17) | 20.4(±3.54) | 28.8(±3.97) | **44.0(±4.36)** |
| **Avg Win%** | 39.9(±0.96) | 40.3(±0.96) | **40.9(±0.96)** | 39.9(±0.96) |

Figure: Average win rate over 5 levels of each game

# Conclusion

General game playing is an exciting, still young but on the verge of maturing topic, which touches upon a broad range of aspects of artificial intelligence. it provides a rich source of interesting and challenging problems for many an AI researcher.

# References I


Perez-Liebana, D., Samothrakis, S., Togelius, J., Lucas, S. M., & Schaul, T. (2016).
General video game AI: Competition, challenges, and opportunities.
In 30th AAAI Conference on Artificial Intelligence, AAAI 2016, 4335–4337.

Perez, D., Samothrakis, S., & Lucas, S. (2014).
Knowledge-based fast evolutionary MCTS for general video game playing.
IEEE Conference on Computatonal Intelligence and Games, CIG, 1–8.

Yannakakis, G. N. (2012).
Game AI revisited.
In CF '12 - Proceedings of the ACM Computing Frontiers Conference, 285–292.

Togelius, J., & Yannakakis, G. N. (2016).
General general game AI.
In IEEE Conference on Computatonal Intelligence and Games, CIG, 0, 1–8.

# References II


Hausknecht, M., Lehman, J., Miikkulainen, R., & Stone, P. (2014).
A neuroevolution approach to general atari game playing.
In IEEE Transactions on Computational Intelligence and AI in Games, 6(4), 355–366.

Naddaf, Y. (2010).
Game-independent AI agents for playing Atari 2600 console games.
In ProQuest Dissertations and Theses, MR60546, 79.

Bellemare, M. G., Veness, J., & Bowling, M. (2012).
Investigating contingency awareness using Atari 2600 games.
In Proceedings of the National Conference on Artificial Intelligence, 2, 864–871.

Perez, D., Samothrakis, S., & Lucas, S. (2014).
Knowledge-based fast evolutionary MCTS for general video game playing.
In IEEE Conference on Computatonal Intelligence and Games, CIG, 1–8.

# References III


Park, H., & Kim, K. J. (2015).
MCTS with influence map for general video game playing.
In 2015 IEEE Conference on Computational Intelligence and Games, CIG 2015 - Proceedings, 534–535.

Sironi, C. F., Liu, J., Perez-Liebana, D., Gaina, R. D., Bravi, I., Lucas, S. M., & Winands, M. H. M. (2018).
Self-adaptive MCTS for General Video Game Playing.
In Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 10784 LNCS(January), 358–375.

Mendes, A., Togelius, J., & Nealen, A. (2016).
Hyper-heuristic general video game playing.
In IEEE Conference on Computatonal Intelligence and Games, CIG, 0, 1–8.

# References IV


Thielscher, M. (2011).

General game playing in AI research and education.

In Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 7006 LNAI, 26–37.

The End