Evolutionary Computation and Games
Julian Togelius, Sebastian Risi, Georgios N. Yannakakis

Who we?
Julian
Georgios
Sebastian

modl.ai

Want to know more?

Evolutionary computation can be used to...

• Play games
• Generate game content (levels etc)
• Generate games
• Model players
• Assist designers
• <your idea here>

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the Owner/Author.

GECCO '20 Companion, July 8–12, 2020, Cancún, Mexico
© 2020 Copyright is held by the owner/author(s).
https://doi.org/10.1145/3377929.3389854
Playing board games

AI applied to games

Playing board games
How can evolution be used to play a game?

Surely, deep Q-learning is the best algorithm for game-playing!
• Planning (requires forward model)
  • Uninformed search (e.g. minimax, breadth-first)
  • Informed search (e.g. A*)
  • Evolutionary algorithms
• Reinforcement learning (requires training time)
  • TD-learning / approximate dynamic programming
  • Evolutionary algorithms
• Supervised learning (requires play traces to learn from)
  • Neural nets, k-nearest neighbors etc
• Random (requires nothing)

How can evolution be used to play a game?

• Evolve an agent that plays the game
  • e.g. through neuroevolution or genetic programming
• Use evolution to play the game (as an action selector)
Evolving Neural Networks

- Direct encodings
  - Evolution strategies / Genetic algorithms
  - NEAT (can evolve arbitrary topologies)
  - Many more …

- Indirect encodings
  - HyperNEAT
  - Compressed weight space
  - Many more …

Why Neuroevolution

- Broad applicability
- Can be used for both supervised and RL problems
- Diversity
- Open-ended learning
- Enables new types of games

NERO: NeuroEvolving Robotic Operatives (Stanley et al. 2005)

- NPCs improve in real time as game is played
- Player can train AI for goal and style of play
- Each AI Unit Has Unique NN
- Supports incremental evolution

EvoCommander

New game mechanics based on brain switching (Jallov et al. 2015)

https://www.youtube.com/watch?v=xWjbCe5Zo8#t=22
Fitness Evaluations in Games

- Co-evolution
- Multiobjective Evolution
- Incremental Evolution

**NE Role: Direct action selection**

**Car racing**

- Driving a car fast requires fine motor control (in both senses)
- Optimizing lap times requires planning
- Overtaking requires adversarial planning
A simple car game

- Walls are solid
- Waypoints must be passed in order
- Fitness: continuous approximation of waypoints passed in 700 time steps

\[ \text{Fitness} = \text{approximation of waypoints passed} \]

\[ \text{Fitness (sd)} \]

<table>
<thead>
<tr>
<th>Track</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fitness (sd)</td>
<td>2.22 (0.09)</td>
<td>2.12 (0.04)</td>
<td>2.06 (0.1)</td>
</tr>
</tbody>
</table>

**Table VI**

- Inputs
  - Six range-finder sensors (evolvable pos.)
  - Waypoint sensor, Speed, Bias
- Networks
  - Standard multi-layer perceptron, 9:6:2
  - Outputs interpreted as thrust/steering

**Algorithm 2: Evolution Strategy(µ,λ,ρ)**

1. **INITIALIZE** (Population, µ + λ individuals)
2. for i=1 to µ do
   3. for j=1 to (µ + λ) do
      4. **EVALUATE** (Population[i])
   5. end
   6. **PERMUTE** (Population)
   7. **SORTONFITNESS** (Population)
   8. for j=µ to (µ + λ) do
      10. **WEIGHTMUTATE** (Population[i])
   11. end
   12. end

Mutation: add Gaussian noise with sd 1 to each connection

Fitness: progress around the track
Example video

Evolved with 50+50 ES, 100 Generations

Choose your inputs (+their representation)

- Using third-person inputs (cartesian inputs) seems not to work
- Either range-finders or waypoint sensor can be taken away, but some fitness lost
- A little bit of noise is not a problem, actually it's desirable
- Adding extra inputs (while keeping core inputs) can reduce evolvability drastically!

Generalization and specialization

- A controller evolved for one track does not necessarily perform well on other tracks
- How do we achieve more general game-playing skills?
- Is there a tradeoff between generality and performance?
Incremental evolution

- Introduced by Gomez & Mikkulainen (1997)
- Change the fitness function $f$ (to make it more demanding) as soon as a certain fitness is achieved
- In this case, add new tracks to $f$ as soon as the controller can drive 1.5 rounds on all tracks currently in $f$

Video: navigating a complex track

Observations

- Controllers evolved for specific tracks perform poorly on other tracks
- General controllers, that can drive almost any track, can be incrementally evolved
- Starting from a general controller, a controller can be further evolved for specialization on a particular track
  - drive faster than the general controller
  - works even when evolution from scratch did not work!
Two cars on a track

- Two car with solo-evolved controllers on one track: disaster
- they don’t even see each other!
- How do we train controllers that take other drivers into account? (avoiding collisions or using them to their advantage)
- Solution: car sensors (rangefinders, like the wall sensors) and *competitive coevolution*

Competitive coevolution

- The fitness function evaluates at least two individuals
- One individual's success is *adversely* affected by the other's (directly or indirectly)
- Very potent, but seldom straightforward; e.g. Hillis (1991), Rosin and Belew (1996)

Competitive coevolution

- Standard 15+15 ES; each individual is evaluated through testing against the current best individual in the population
- Fitness function a mix of...
  - Absolute fitness: progress in \( n \) time steps
  - Relative fitness: distance ahead of or behind the other car after \( n \) time steps

Video: absolute fitness
Open Challenges: NE in Games

- Reaching Record-beating Performance
- Combining evolution with other learning methods
- Learning from high-dimensional/raw data
- General video game playing
- Combining NE with life-long learning
- Competitive and cooperative coevolution
- Fast and reliable methods for commercial games

Emerging Trends – Hybrid Methods

Alvernaz and Togelius, 2017

Ha and Schmidhuber, 2018

Voit et al., 2018
Using evolution to plan?

- Some games have extremely high branching factor
  - Chess: 35
  - Go: 350
  - Civilization/StarCraft: say you have ten units, which can each take one of ten actions…
- Tree search cannot even get past the first ply
- One solution: treat the whole plan as a sequence of actions, the value of the final state as fitness…

Hero Academy

Enormous branching factor beats MCTS

<table>
<thead>
<tr>
<th></th>
<th>Random</th>
<th>Greedy Action</th>
<th>Greedy Turn</th>
<th>MCTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Greedy Action</td>
<td>100%</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Greedy Turn</td>
<td>100%</td>
<td>64.0%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MCTS</td>
<td>100%</td>
<td>48.5%</td>
<td>22.0%</td>
<td></td>
</tr>
</tbody>
</table>

Online Evolutionary Planning

- Evolve the set of actions to take each turn
- Chromosome is a sequence of five actions
- Simple evolutionary algorithm:
  - Population size of 100, 50% elitism, random selection of parents, uniform crossover, 10% mutation rate
Results: wow

- ~10,000 unique outcomes evaluated each turn (6 seconds)
- ~3,500 generations each turn on average


Procedural content generation in games

Why generate game content?

- To replace the human? (Saving time and money…)
- To assist the human designer?
- To make new types of games possible?
- To go beyond human creativity
- To really understand design
Search-based PCG

- Use evolutionary computation to search the design space for good artifacts (e.g. levels)
- Technically, we could use other stochastic search / optimization algorithms
- Major issues:
  - Representing the content
  - Devising a good evaluation / fitness function


Petalz Social Facebook Game
based on PCG through NE

Sebastian Risi, Joel Lehman, David D’Ambrosio, Ryan Hall, Kenneth Stanley, AIIDE 2012, TCIAIG 2015

Generating Flower Images and Shapes

Sebastian Risi, Joel Lehman, David D’Ambrosio, Ryan Hall, Kenneth Stanley, AIIDE 2012, TCIAIG 2015
Generating Flower Images and Shapes

Flower Evolution: Pollinating a Flower

Planting the Offspring
Crosspollination Also Possible

Hybrid Methods - Latent Variable Evolution (LVE)

- A learned compact genotype-to-phenotype mapping ⇒ robust mutations
- Applicable to variety of other domains

Bontrager, Togelius, Memon 2017
Bontrager, Lin, Togelius, Risi, 2018

Crosspollination

Generative and Adversarial Networks (GANs)

Goodfellow 2014

Radford et al. 2015

https://deeplearning4j.org/generative-adversarial-network
Evolving Mario Levels in the Latent Space of a Deep Convolutional Generative Adversarial Network
Voiz, Schrum, Liu, Lucas, Smith, Risi, GECCO 2018

**Approach – Phase II**

**GAN Training**

173 training images of size 28x14

**Level Representation**

<table>
<thead>
<tr>
<th>Tile type</th>
<th>Symbol</th>
<th>Identity</th>
<th>Visualization</th>
</tr>
</thead>
<tbody>
<tr>
<td>Solid/Ground</td>
<td>X</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Breakable</td>
<td>S</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Empty (passable)</td>
<td>-</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>Full question block</td>
<td>?</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>Empty question block</td>
<td>Q</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>Enemy</td>
<td>E</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>Top-left pipe</td>
<td>&lt;</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>Top-right pipe</td>
<td>&gt;</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>Left pipe</td>
<td>[</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>Right pipe</td>
<td>]</td>
<td>9</td>
<td></td>
</tr>
</tbody>
</table>

GAN changes:
- One-hot encoding
- ReLU activation function for output layer
- Argmax to determine tile type
CMA-ES Experiments

• Representation-based testing:
  – Optimize for certain number of ground titles
    \[ F_{ground} = \sqrt{(g - t)^2} \]
  – Increasing difficulty (less ground, more enemies)

• Agent-based testing:
  A* Mario agent by Baumgarten
  Fitness = %playable + #jumps

⇒ Trained GAN representation displays locality
Results

![Game AI Play Games Generate Content Model Players](image)

Multiobjective exploration of the Starcraft map space. In Computational Intelligence and Games (CIG), IEEE Conference on, pp. 265–272, 2010.

Procedural FPS Level Generation


Deep Learning Meets Novelty Search

Constrained Novelty Search


From Novelty Search to **Surprise Search**


**Surprise Search for Problem Solving**


**Surprise for QD**

- Novelty-Surprise Search: a robust and efficient divergent search algorithm
  - Maze navigation
  - Robot morphology evolution
- Surprise for quality diversity
  - Combined with local competition is highly advantageous


How – In a Nutshell

Experience: Labels are Key!

To sum it up: **Don’t** do this!

- Wasteful Info due to
- Scale-bias
- Personal-bias
- Labels are **NOT** numbers
- High inconsistency (randomness)
- ...

**The ordinal (relative) approach**


Arousal

Happier

Valence (negative - positive)

Do this instead

- I like **Julian’s** style more/less than **Georgios’s** style
- I like them both equally
- I like neither

- You gain on
  - Reliability
  - Validity
  - Generality
Modeling Player Experience

Supervised learning for modelling experience

- Nominal values
  - Julian is frustrated
- Numerical values
  - Julian is 0.86 frustrated
- Ordinal values
  - Georgios is more frustrated than Julian

Which Training Method?

Preference Learning

- Preference learning is inspired by and built upon humans’ limited ability to express their preferences directly in terms of a specific (subjective) value function
- Our inability is mainly due to the
  - subjective nature of a preference
  - cognitive load for assigning specific values to each one of the options
- It is more natural to express preferences about a number of options; and this is what we end up doing normally.

(Deep) Preference Learning with BP

- Error function maximizes the distance between the output for the preferred sample \((d^A)\) and the output for the non preferred sample \((d^B)\)

\[
E = \max(0, 1 - (d^A - d^B)) \quad \frac{\partial E}{\partial w_{ij}} = \begin{cases} 
\frac{\partial d^A}{\partial w_{ij}} + \frac{\partial d^B}{\partial w_{ij}}, & \text{if } d^A - d^B < 1 \\
0, & \text{otherwise}
\end{cases}
\]


(Deep) Preference Learning beyond BP

- Learning from pairs of preferences can be implemented in most supervised learning methods by adapting the error/fitness function
  - Neuroevolution
    - Fitness that rewards match of pairs
  - Rank-based ANN (RankNet)
  - SVMs (RankSVM)
  - Decision Trees
  - ...

An Open-Source Preference Learning Toolbox


https://sourceforge.net/projects/pl-toolbox/
Emotionally Adaptive Cameras
Yannakakis, Martinez, Jhala, Towards Affective Camera Control in Games, UMUAI, 2010

General Models of Affect
Camilleri, Yannakakis and Liapis, Towards General Models of Player Affect, ACII, 2017

You have a Player Model... so what?
Experience-driven PCG

Model you!
Design your Game!
Game AI

EDPCG: What is it?

“A framework for personalised generation of content in human computer interaction (in particular in games). It views (game) content as the building block of user (player) experience”


Experience-Driven Level Generation in Super Mario Bros

Platformer Experience Dataset

http://ped.institutedigitalgames.com/
Reframing Mario

Game Design for Agent Believability
Camilleri, Yannakakis and Dingli, Platformer Level Design for Player Believability, IEEE CIG, 2016

Game Design for Agent Believability
Camilleri, Yannakakis and Dingli, Platformer Level Design for Player Believability, IEEE CIG, 2016

Player Modeling Beyond Supervised Learning
**Designer Modeling: Procedural Strategy Map Design**

---

**Procedural Personas**
- Given utilities (rewards) show me believable gameplay
- Useful for human-standard game testing
- RL
  - MCTS
  - Neuroevolution
  - ...
- Inverse RL


---

**Orchestration**

“Games: the final frontier for AI?”

“AI: the next step for Games!”

Get Involved!

IEEETRANSACTIOSON
GAMES

A PUBLICATIONOF THEIEEECOMPUTATIONALINTELLIGENCESOCIETY,
THEIEEESENSORS COUNCIL, AND THEIEEE CONSUMER ELECTRONICS SOCIETY

http://cis.ieee.org/ieeetransactions-on-games.html

Thank you!
gameai book.org