Evolutionary Computation for Creativity and Intelligence

By Darwin Johnson, Alice Quintanilla, and Isabel Tweraser
Introduction to NEAT

- Stands for NeuroEvolution of Augmenting Topologies (NEAT)
- Evolves an Artificial Neural Network of nodes (simple artificial “brain”)
  - Generates population of ANNs to solve a problem (usually ineffective at first)
  - Best performing ANNs continue to the next generation and produce children
  - Mutates the ANNs to change the behavior
    - Alters the weight of an existing link between two nodes
    - Creates a new link between nodes
    - Creates new nodes along existing links
Mutations

- **Alter Link Weight**
  - Alters the influence of an existing link
  - May increase or decrease the weight

- **Add Link**
  - Adds a link between two existing nodes
  - Creates influence from one node to another

- **Add Node**
  - Creates a new node along an existing link
  - Adds new influence to the final output
  - Creates new location for links to connect to
Compositional Pattern Producing Networks (CPPNS)

- Variant of ANN with variety of activation functions in its nodes
- Activation functions create patterns reminiscent of features of natural life:
  - Repetition
  - Symmetry
  - Variation
- Can be used to generate interesting images and sounds
saw(1*tanh(x*0.5+(-0.75)*sin(x)) + (0.7*sin(x*(-0.25)+0.3*sin(x))))
Evolving Music and Sounds

- Use a CPPN to generate an amplitude wave
- Can be displayed and played as a sound wave

- Two extensions:
  - *Breedesizer* - Evolves sounds that can be played with different frequencies/notes*
  - *Remixbreeder* - Takes in a song and outputs a remixed version

Breedesizer Interface
Board Games

- Common tests for Artificial Intelligence
- Tic-Tac-Toe, Checkers, Othello, and more
- Several Opponent choices to create an Agent:
  - Static Opponent
  - Co-Evolution
- Evolve board evaluation functions
  - Board state evaluated by ANN
  - Move with highest output selected
Board Game Opponents

Static Opponent

Agent is evolved against a non-evolving agent.

- Used as a Benchmark
  - Easier to compare against
  - Can be considered a “goal” to reach
- Agents evolve to beat this specific Opponent
  - May not be able to beat other opponents
  - Not necessarily “good” agents

Co-Evolution

Agents are evolved against each other.

- Agents evolve as a group
  - Fitness depends on other agents in population
  - Should learn general intelligent behavior
- More difficult to benchmark
  - Unable to have a consistent opponent
  - Emergence of unusual weaknesses possible

Same Opponent

Evolved Population

VS

Best Individuals Selected

Matches from Population

VS

Best Individuals Selected
**Evaluation of Game States**

Game Trees - A series of branching game states

- Created from all possible sequences of moves in a board game
- Evolved ANNs evaluate move sequences to determine the best current action
- Tree-Search several board states ahead: focus on long-term outcomes
- Searches a limited number of states due to time limit
- Several tree search algorithms exist
  - Monte-Carlo
  - Minimax
  - Minimax with alpha-beta pruning
Applying tree-search: MicroRTS

- RTS: Real-time strategy
  - Players act simultaneously
  - Actions cost time
  - Large branching factor
- MicroRTS
  - Much simpler than real RTS
  - Developed as AI benchmark
    - Generic unit classes
  - Forward simulation
    - Know all possible future states
    - Tree-search
  - Adjustable size
  - International AI competition
- Using NN to evaluate game states
What does it mean to “evaluate a state” in this domain?

- Units’ locations
- How many of each unit type
- Available resources
- Remaining base health
- Etc...

![Game State](image)

Evaluating a state involves:
- Inputting the current state into a neural network.
- The neural network processes the state to output a score.
- The highest scoring state is selected using argmax.

![Neural Network Diagram](image)
Evolved Agent in action!

● Blue player is evolved NN
  ○ Evolved over night
  ○ Uns suited for larger maps
  ○ Video shows its best match
  ○ Performance from 21 gens

● Red player is a simple AI
  ○ Random behavior
  ○ Biased towards performing a predetermined list of actions
  ○ Not particularly hard to beat

● Future work
  ○ Coevolution
  ○ Beating harder opponents
  ○ Evolve for longer
Acknowledgements

Southwestern University

Dr. Jacob Schrum

Past SCOPE students: Gabriela Gonzalez, Alex Rollins, & Lauren Gillespie

Santiago Ontañón (MicroRTS developer)

Questions?