ROB 538: Multiagent Systems

Week 3, Lecture 2:

Cooperative Coevolution and Visualizing Rewards

Reading:

Announcements:
HW 2 due: 10/17
Project Presentations: 11/30: 8-noon

Evolutionary Algorithms

- Population based search
- Evolve policies (neural networks)

- But what if you have many agents?
- Many populations?
- How does that work?
Standard Evolutionary Algorithm

- Initialize a population of $k$ policies
  - Retain $k$ best policies
  - Select one policy and mutate it to create $k+1$ policies
  - Assess performance and assign fitness to new policy
  - Use new policy

Cooperative Coevolutionary Algorithm

- Initialize $M$ populations of $k$ policies
  - Retain $k$ best performing policies of each population
  - Mutate each to create $M$ populations of $2k$ policies
  - Assess team performance and assign fitness to team members
  - Randomly select one policy from each population to create team $T_j$
Cooperative Coevolutionary Algorithm

Population 1

Population 2

Population n

Fitness Assignment

System Performance

System

Team

Cooperative Coevolutionary Algorithm

Population 1

Population 2

Population n

Fitness Assignment

Team

Cooperative Coevolutionary Algorithm
Cooperative Coevolutionary Algorithm

- Fitness of an agent is a function of two things:
  - The agent’s policy
  - How the collaborating agents act

- Fitness assignment in cooperative coevolutionary algorithms is very context-dependent and subjective

- Credit assignment problem extremely difficult to solve
  - Fitness function shaping, including:
    § hall of fame
    § leniency
    § difference evaluation functions

Hall of Fame for Fitness Assignment

Population 1  →  Fitness Assignment  →  System Performance

Population 2  →  Best

Population n  →  Best

Team  →  System
Hall of Fame for Fitness Assignment

Population

Best

System Performance

 System

Population

Best

Team

Fitness Assignment

Hall of Fame Summary

• Designed for competitive coevolution:
  
  test agents against best opponents in addition to standard fitness evaluation

• Bias search towards agents which can beat current known best
Leniency for Fitness Assignment

Population 1

Population 2

Population n

System Performance

Team

System

Fitness Assignment = max \{G_1, G_2, G_3, \ldots, G_N\}
Leniency Summary

- *Leniency* tests an agent with multiple sets of collaborators
- Best system evaluation score is assigned as fitness

An agent won’t get a bad fitness assignment because it happens to get paired with bad teammates

Difference Evaluations for Fitness Assignment
Difference Evaluations for Fitness Assignment

• Find agent’s contribution to system evaluation
• One team, many fitness evaluations

But Wait …

• Leniency ensures you don’t get penalized for a bad team
• Hall of fame tries to create the best team for you
• Difference evaluations give agent specific evaluation
• Why not combine them?
Hall of Fame and Difference Evaluations

Population 1
Fitness Assignment
Best
System Performance
$D_i(z)$
System

Leniency and Difference Evaluations

Population 1
Population 2
Population n
Team
System Performance
System

Team
Leniency and Difference Evaluations

Population 1

Fitness Assignment = \( \max \{ D_1^1, D_1^2, D_1^3, \ldots, D_1^k \} \)

\[
\max \{ D_N^1, D_N^2, D_N^3, \ldots, D_N^k \}
\]

Population 2

Population n

D_1^1 \quad D_1^2 \quad D_1^k

D_N^1 \quad D_N^2 \quad D_N^k

How Does It All Work: Rover Domain

System Performance vs. Generation

G(x)

Generation
How Does It All Work: Rover Domain

![Graph showing system performance over generations.](image_url)
How Does It All Work: Rover Domain

![Graph showing system performance over generations for different conditions, including G(z), D(z), Leniency, HOF, and Lenient D(z).]
How Does It All Work: Rover Domain

How Does It All Work: Scaling
Computational Cost Analysis

- The fitness assignment operators all have different computational costs
  - System evaluation function: 1 call to $G(z)$
  - Difference evaluation function: $n+1$ calls to $G(z)$
  - Leniency and difference evaluation: $nk+n$ calls to $G(z)$
  - Hall of fame and difference evaluation: $2n+1$ calls to $G(z)$

- Need to look at performance as a function of computational cost
Computational Cost Analysis

Questions?
Recall General Solution

Difference Objective:

\[ D_i(z) = G(z) - G(z - i + c_i) \]

- \( D_i \) is aligned with \( G \)
- \( G(z_i + c_i) \) is independent of \( i \)
- \( D_i \) has cleaner signal than \( G \)
- \( G(z_i + c_i) \) removes noise

• How do we test that?
  - Alignment in a real domain?
  - Does it really have a “cleaner” signal?

Rover Coordination

- Rovers observe points of interest (POIs)
  - POIs vary in value, time and place
  - Get more information closer to POI
  - Only primary observation counts
- Learning problem
  - Rovers learn in single trial (non-episodic)
    - Dynamic: POIs appear/disappear
  - Rovers reset at regular intervals (episodic)
    - Static: POIs the same in each episode
    - Dynamic: POIs different in each episode

\[ G = \sum_i \sum_j \frac{V_j}{\min_i \delta(L_{ij}, L_{i,1})} \]

\[ \delta(x, y) = \min \{ ||x - y||^2, d^2 \} \]
### Objective Functions

\[
G = \sum_t \sum_j \frac{V_j}{\min_{i'} \delta(L_{j, t}, L_{i', t})} \quad \text{Global (Fully Factored, Low Learnability)}
\]

\[
P_i = \sum_t \sum_j \frac{V_j}{\delta(L_{j, t}, L_{i, t})} \quad \text{Selfish / Individual (Low Factoredness, } \infty \text{ Learnability)}
\]

\[
D_i = \sum_t \left[ \sum_j \frac{V_j}{\min_{i'} \delta(L_{j, t}, L_{i', t})} - \sum_j \frac{V_j}{\min_{i'} \neq i \delta(L_{j, t}, L_{i', t})} \right] \quad \text{Difference (High Factoredness, High Learnability)}
\]

### Agent State

- Four rover sensors
- Four point of interest sensors
- Each sensor returns a sum over a quadrant based on:
  - Number of observations
  - Distance of observation
  - Value of observation (for POIs only)
Agent State Projection

Factoredness Computation:
- Select action
- Receive system reward
- Compute agent reward
- Check alignment

Analyzed Rewards

- \( P_i \): Sum of POI values observed by agent \( i \)
- \( G_i \): Sum of POI values observed by all agents
- \( D_i \): Sum of POI values observed by agent \( i \) that would have gone unobserved by other agents
- \( D_i(PO) \): \( D_i \) with rovers communication restricted to distance they can travel in one step (3% of space)
Project on Problem Domain

Questions?