CS344M
Autonomous Multiagent Systems

Todd Hester

Department or Computer Science
The University of Texas at Austin

Good Afternoon, Colleagues

Are there any questions?

Logistics

- Readings

Logistics

- Readings
  - Specify which papers you read!

Logistics

- Readings
  - Specify which papers you read!
  - 2 case studies and 1 TDP

Logistics

- Readings
  - Specify which papers you read!
  - 2 case studies and 1 TDP
- How to read a research paper

Logistics

- Readings
  - Specify which papers you read!
  - 2 case studies and 1 TDP

- How to read a research paper
  - Some have too few details...

Logistics

- Readings
  - Specify which papers you read!
  - 2 case studies and 1 TDP
- How to read a research paper
  - Some have too few details...
  - Others have too many.

Logistics

- Readings
  - Specify which papers you read!
  - 2 case studies and 1 TDP
- How to read a research paper
  - Some have too few details...
  - Others have too many.
- Next week's readings posted

Logistics

- Readings
  - Specify which papers you read!
  - 2 case studies and 1 TDP
- How to read a research paper
  - Some have too few details...
  - Others have too many.
- Next week's readings posted
- Use the undergrad writing center!
  - Friday afternoon workshops (3 p.m.)

Overview of the Readings

- *Darwin:* genetic programming approach

Overview of the Readings

- *Darwin:* genetic programming approach
- *Stone and McAllester:* Architecture for action selection

Overview of the Readings

- *Darwin:* genetic programming approach
- *Stone and McAllester:* Architecture for action selection
- *Riley et al:* Coach competition, extracting models

Overview of the Readings

- *Darwin:* genetic programming approach
- *Stone and McAllester:* Architecture for action selection
- *Riley et al:* Coach competition, extracting models
- *Kuhlmann et al:* Learning for coaching

Overview of the Readings

- *Darwin:* genetic programming approach
- *Stone and McAllester:* Architecture for action selection
- *Riley et al:* Coach competition, extracting models
- *Kuhlmann et al:* Learning for coaching
- *Withopf and Riedmiller:* Reinforcement learning

Overview of the Readings

- *Darwin:* genetic programming approach
- *Stone and McAllester:* Architecture for action selection
- *Riley et al:* Coach competition, extracting models
- *Kuhlmann et al:* Learning for coaching
- *Withopf and Riedmiller:* Reinforcement learning
- *MacAlpine et al:* UT Austin Villa 2011

Overview of the Readings

- *Darwin:* genetic programming approach
- *Stone and McAllester:* Architecture for action selection
- *Riley et al:* Coach competition, extracting models
- *Kuhlmann et al:* Learning for coaching
- *Withopf and Riedmiller:* Reinforcement learning
- *MacAlpine et al:* UT Austin Villa 2011
- *Barrett et al:* SPL Kicking strategy

Evolutionary Computation

- Motivated by biological evolution: GA, GP

Evolutionary Computation

- Motivated by biological evolution: GA, GP
- Search through a space

Evolutionary Computation

- Motivated by biological evolution: GA, GP
- Search through a space
  - Need a **representation, fitness function**
  - Probabilistically apply search operators to set of points in search space

Evolutionary Computation

- Motivated by biological evolution: GA, GP
- Search through a space
  - Need a **representation, fitness function**
  - Probabilistically apply search operators to set of points in search space
- Randomized, parallel hill-climbing through space

Evolutionary Computation

- Motivated by biological evolution: GA, GP
- Search through a space
  - Need a **representation, fitness function**
  - Probabilistically apply search operators to set of points in search space
- Randomized, parallel hill-climbing through space
- Learning is an optimization problem (fitness)

Evolutionary Computation

- Motivated by biological evolution: GA, GP
- Search through a space
  - Need a **representation, fitness function**
  - Probabilistically apply search operators to set of points in search space
- Randomized, parallel hill-climbing through space
- Learning is an optimization problem (fitness)

Some slides from *Machine Learning* (Mitchell, 1997)

Darwin United

- More ambitious follow-up to Luke, 97 (made 2nd round)

Darwin United

- More ambitious follow-up to Luke, 97 (made 2nd round)
- Motivated in part by Peter's detailed team construction

Darwin United

- More ambitious follow-up to Luke, 97 (made 2nd round)
- Motivated in part by Peter's detailed team construction
- Evolves whole teams — lexicographic fitness function

Darwin United

- More ambitious follow-up to Luke, 97 (made 2nd round)
- Motivated in part by Peter's detailed team construction
- Evolves whole teams — lexicographic fitness function
- Evolved on huge (at the time) hypercube

Darwin United

- More ambitious follow-up to Luke, 97 (made 2nd round)
- Motivated in part by Peter's detailed team construction
- Evolves whole teams — lexicographic fitness function
- Evolved on huge (at the time) hypercube
- Lots of spinning, but figured out dribbling, offsides

Darwin United

- More ambitious follow-up to Luke, 97 (made 2nd round)
- Motivated in part by Peter's detailed team construction
- Evolves whole teams — lexicographic fitness function
- Evolved on huge (at the time) hypercube
- Lots of spinning, but figured out dribbling, offsides
- 1-1-1 record. Tied a good team, but didn't advance

Darwin United

- More ambitious follow-up to Luke, 97 (made 2nd round)
- Motivated in part by Peter's detailed team construction
- Evolves whole teams — lexicographic fitness function
- Evolved on huge (at the time) hypercube
- Lots of spinning, but figured out dribbling, offsides
- 1-1-1 record. Tied a good team, but didn't advance
- Success of the method, but not pursued

Architecture for Action Selection

- (other slides, video)

Architecture for Action Selection

- (other slides, video)
- downsides

Architecture for Action Selection

- (other slides, video)
- downsides
- Keepaway

Coaching

- Learn best strategy to play a fixed team

Coaching

- Learn best strategy to play a fixed team
- Give high level advice to players at low frequency

Coaching

- Learn best strategy to play a fixed team
- Give high level advice to players at low frequency
- Focus on learning formations

Coaching

- Learn best strategy to play a fixed team
- Give high level advice to players at low frequency
- Focus on learning formations
- Learn when successful teams passed/kicked

Coaching

- Learn best strategy to play a fixed team
- Give high level advice to players at low frequency
- Focus on learning formations
- Learn when successful teams passed/kicked
- Learn when opponent will pass and try to block

Coaching

- Learn best strategy to play a fixed team
- Give high level advice to players at low frequency
- Focus on learning formations
- Learn when successful teams passed/kicked
- Learn when opponent will pass and try to block
- What if players switch roles?

Coaching

- Learn best strategy to play a fixed team
- Give high level advice to players at low frequency
- Focus on learning formations
- Learn when successful teams passed/kicked
- Learn when opponent will pass and try to block
- What if players switch roles?
- Why just imitate another team?

Coaching

- Learn best strategy to play a fixed team
- Give high level advice to players at low frequency
- Focus on learning formations
- Learn when successful teams passed/kicked
- Learn when opponent will pass and try to block
- What if players switch roles?
- Why just imitate another team?
- Other slides

Reinforcement Learning

- RL Slides

Reinforcement Learning

- RL Slides
- Extend to grid soccer

Reinforcement Learning

- RL Slides
- Extend to grid soccer
- Large state space, joint actions

Reinforcement Learning

- RL Slides
- Extend to grid soccer
- Large state space, joint actions

UT Austin Villa 2011

- Other slides

UT Austin Villa 2011

- Other slides
- Why not use CMA-ES on role positions as well?

UT Austin Villa 2011

- Other slides
- Why not use CMA-ES on role positions as well?
- Changes for 2012?

Kicking Under Uncertainty

- Used by our SPL team

Kicking Under Uncertainty

- Used by our SPL team

- Kick engine to kick at various distances/headings

Kicking Under Uncertainty

- Used by our SPL team
- Kick engine to kick at various distances/headings
- Adjust to seen ball location

Kicking Under Uncertainty

- Used by our SPL team
- Kick engine to kick at various distances/headings
- Adjust to seen ball location
- Select first kick that moves ball up field

Kicking Under Uncertainty

- Used by our SPL team
- Kick engine to kick at various distances/headings
- Adjust to seen ball location
- Select first kick that moves ball up field
- Figure

Kicking Under Uncertainty

- Used by our SPL team
- Kick engine to kick at various distances/headings
- Adjust to seen ball location
- Select first kick that moves ball up field
- Figure
- Emphasis on quickness

Kicking Under Uncertainty

- Used by our SPL team
- Kick engine to kick at various distances/headings
- Adjust to seen ball location
- Select first kick that moves ball up field
- Figure
- Emphasis on quickness
- Now: Better model of opponents -> Know if we have more time

Kicking Under Uncertainty

- Used by our SPL team
- Kick engine to kick at various distances/headings
- Adjust to seen ball location
- Select first kick that moves ball up field
- Figure
- Emphasis on quickness
- Now: Better model of opponents -> Know if we have more time

Learning Commentary

- David Chen and Ray Mooney

Coordination Graphs

- n agents, each choose an action A_i

Coordination Graphs

- n agents, each choose an action A_i
- $A = A_1 \times \ldots \times A_n$

Coordination Graphs

- n agents, each choose an action A_i
- $A = A_1 \times \ldots \times A_n$
- $R_i(A) \mapsto \mathbb{R}$

Coordination Graphs

- n agents, each choose an action A_i
- $A = A_1 \times \ldots \times A_n$
- $R_i(A) \mapsto \mathbb{R}$
- Coordination problem: $R_1 = \ldots = R_n = R$

Coordination Graphs

- n agents, each choose an action A_i
- $A = A_1 \times \ldots \times A_n$
- $R_i(A) \mapsto \mathbb{R}$
- Coordination problem: $R_1 = \ldots = R_n = R$
- Nash equilibrium: no agent could do better given what others are doing.

Coordination Graphs

- n agents, each choose an action A_i
- $A = A_1 \times \ldots \times A_n$
- $R_i(A) \mapsto \mathbb{R}$
- Coordination problem: $R_1 = \ldots = R_n = R$
- Nash equilibrium: no agent could do better given what others are doing.
- May be more than one (chicken)

Example from the paper

- Understand the rule syntax

Example from the paper

- Understand the rule syntax
- Form the coordination graph

Example from the paper

- Understand the rule syntax
- Form the coordination graph
- First eliminate rules based on context

Example from the paper

- Understand the rule syntax
- Form the coordination graph
- First eliminate rules based on context
- What does it mean for G_3 to collect all relevant rules?

Example from the paper

- Understand the rule syntax
- Form the coordination graph
- First eliminate rules based on context
- What does it mean for G_3 to collect all relevant rules?
- What does it mean for G_3 to maximize over all actions of a_1 and a_2?

Example from the paper

- Understand the rule syntax
- Form the coordination graph
- First eliminate rules based on context
- What does it mean for G_3 to collect all relevant rules?
- What does it mean for G_3 to maximize over all actions of a_1 and a_2?
- How are the results propagated back?

Example from the paper

- Understand the rule syntax
- Form the coordination graph
- First eliminate rules based on context
- What does it mean for G_3 to collect all relevant rules?
- What does it mean for G_3 to maximize over all actions of a_1 and a_2?
- How are the results propagated back?
- Let's try again with G_1 eliminated first

Application to soccer

- Make the world discrete by assigning roles, using high-level predicates

Application to soccer

- Make the world discrete by assigning roles, using high-level predicates
- Assume global state information

Application to soccer

- Make the world discrete by assigning roles, using high-level predicates

- Assume global state information

- Finds pass sequences and starts players moving ahead of time.

Application to soccer

- Make the world discrete by assigning roles, using high-level predicates
- Assume global state information
- Finds pass sequences and starts players moving ahead of time.
- Note the results: with and without coordination.

Reactive Deliberation

- A hybrid approach
- Executor: carry out reactive behaviors
- Deliberator: evaluate possible high-level schema with parameters; generate bids
- Deliberator takes time, but something keeps happening always.
- In effect: deliberator commits to schema for some time