Autonomous Learning Agents: Layered Learning and Ad Hoc Teamwork

Peter Stone

Learning Agents Research Group (LARG)
Department of Computer Science
The University of Texas at Austin

(Also, Cogitai Inc.)

Thanks to ACM/SIGAI and the AAMAS PCs
A Goal of AI and Robotics
A Goal of AI and Robotics

Robust, **fully autonomous** agents in the real world
A Goal of AI and Robotics

Robust, fully autonomous agents in the real world

How?
A Goal of AI and Robotics

Robust, fully autonomous agents in the real world

How?
- Build complete agents to perform increasingly complex tasks
A Goal of AI and Robotics

Robust, **fully autonomous** agents in the real world

**How?**

- Build complete agents to perform increasingly complex tasks
- Complete agents: sense, decide, and act — closed loop
A Goal of AI and Robotics

Robust, fully autonomous agents in the real world

How?

- Build complete agents to perform increasingly complex tasks
  Complete agents: sense, decide, and act — closed loop

- Drives research on component algorithms, theory
A Goal of AI and Robotics

Robust, **fully autonomous** agents in the real world

**How?**

- Build complete agents to perform increasingly complex tasks
  Complete agents: sense, decide, and act — closed loop

- Drives research on component algorithms, theory
  - Improve from experience  \((\text{Machine learning})\)
A Goal of AI and Robotics

Robust, fully autonomous agents in the real world

How?

- Build complete agents to perform increasingly complex tasks
  Complete agents: sense, decide, and act — closed loop

- Drives research on component algorithms, theory
  - Improve from experience (Machine learning)
  - Interact with other agents (Multiagent systems)
Research Question

To what degree can autonomous intelligent agents learn in the presence of teammates and/or adversaries in real-time, dynamic domains?
Research Question

To what degree can autonomous intelligent agents learn in the presence of teammates and/or adversaries in real-time, dynamic domains?

Research Areas

- Autonomous agents
Research Question

To what degree can autonomous intelligent agents learn in the presence of teammates and/or adversaries in real-time, dynamic domains?

Research Areas

- Autonomous agents
- Machine learning
Research Question

To what degree can autonomous intelligent agents learn in the presence of teammates and/or adversaries in real-time, dynamic domains?

Research Areas

- Autonomous agents
- Machine learning
  - Reinforcement learning
Research Question

To what degree can autonomous intelligent agents learn in the presence of teammates and/or adversaries in real-time, dynamic domains?

Research Areas

- Autonomous agents
- Machine learning
  - Reinforcement learning
- Multiagent systems
Research Question

To what degree can autonomous intelligent agents learn in the presence of teammates and/or adversaries in real-time, dynamic domains?

Research Areas

- Autonomous agents
- Machine learning
  - Reinforcement learning
- Multiagent systems
  - Trading agents
Research Question

To what degree can autonomous intelligent agents learn in the presence of teammates and/or adversaries in real-time, dynamic domains?

Research Areas

- Autonomous agents
- Machine learning
  - Reinforcement learning
- Multiagent systems
  - Trading agents
- Robotics
Research Question

To what degree can autonomous intelligent agents learn in the presence of teammates and/or adversaries in real-time, dynamic domains?

Research Areas

- Autonomous agents
- Machine learning
  - Reinforcement learning
- Multiagent systems
  - Trading agents
- Robotics
Learning in practice:
Autonomous Learning Agents

- **Learning in practice:** Layered learning
Autonomous Learning Agents

- **Learning in practice:** Layered learning
- **Learning in MAS:**
Autonomous Learning Agents

- **Learning in practice**: Layered learning
- **Learning in MAS**: Ad hoc teamwork
RoboCup Soccer
RoboCup Soccer

- Grand challenge: beat World Cup champions by 2050
RoboCup Soccer

- Grand challenge: beat World Cup champions by 2050
- Still in relatively early stages
RoboCup Soccer

- Grand challenge: beat World Cup champions by 2050
- Still in relatively early stages
- Many virtues as a challenge problem:
  - Incremental challenges, closed loop at each stage
  - Robot design to multi-robot systems
  - Relatively easy entry
  - Inspiring to many
RoboCup Soccer

- Grand challenge: beat World Cup champions by 2050
- Still in relatively early stages
- Many virtues as a challenge problem:
  - Incremental challenges, closed loop at each stage
  - Robot design to multi-robot systems
  - Relatively easy entry
  - Inspiring to many

- Visible progress
RoboCup Soccer

- Grand challenge: beat World Cup champions by 2050
- Still in relatively early stages
- Many virtues as a challenge problem:
  - Incremental challenges, closed loop at each stage
  - Robot design to multi-robot systems
  - Relatively easy entry
  - Inspiring to many
- Visible progress

“Good problems . . . produce good science” [Cohen, ’04]
Layered Learning

• For domains too complex for tractably mapping state features $S \mapsto O$

• Hierarchical subtask decomposition given: $\{L_1, L_2, \ldots, L_n\}$
Layered Learning

- For domains too complex for tractably mapping state features $S \rightarrow$ outputs $O$
- Hierarchical subtask decomposition given: $\{L_1, L_2, \ldots, L_n\}$
- Machine learning: exploit data to train, adapt
Layered Learning

- For domains too complex for tractably mapping state features $S \rightarrow O$
- Hierarchical subtask decomposition given: $\{L_1, L_2, \ldots, L_n\}$
- Machine learning: exploit data to train, adapt
- Learning in one layer feeds into next layer
Layered Learning

- For domains too complex for tractably mapping state features $S \rightarrow$ outputs $O$
- Hierarchical subtask decomposition given: $\{L_1, L_2, \ldots, L_n\}$
- Machine learning: exploit data to train, adapt
- Learning in one layer feeds into next layer

High Level Goals

- Adversarial Behaviors
- Team Behaviors
- Multi-Agent Behaviors
- Individual Behaviors

World State

Environment

Machine Learning Opportunities
Layered Learning in Practice

First applied in *simulated* robot soccer [Stone & Veloso, ’97]

<table>
<thead>
<tr>
<th>Strategic Level</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>$L_1$</td>
<td>individual</td>
</tr>
<tr>
<td>$L_2$</td>
<td>multiagent</td>
</tr>
<tr>
<td>$L_3$</td>
<td><strong>team</strong></td>
</tr>
<tr>
<td></td>
<td>ball interception</td>
</tr>
<tr>
<td></td>
<td>pass evaluation</td>
</tr>
<tr>
<td></td>
<td>pass selection</td>
</tr>
</tbody>
</table>
Layered Learning in Practice

First applied in *simulated* robot soccer [Stone & Veloso, ’97]

<table>
<thead>
<tr>
<th></th>
<th>Strategic Level</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>$L_1$</td>
<td>individual</td>
<td>ball interception</td>
</tr>
<tr>
<td>$L_2$</td>
<td>multiagent</td>
<td>pass evaluation</td>
</tr>
<tr>
<td>$L_3$</td>
<td><strong>team</strong></td>
<td>pass selection</td>
</tr>
</tbody>
</table>

Later applied on *real robots* [Stone, Kohl, & Fidelman, ’06]

<table>
<thead>
<tr>
<th></th>
<th>Strategic Level</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>$L_1$</td>
<td>individual</td>
<td>fast walking</td>
</tr>
<tr>
<td>$L_2$</td>
<td>individual</td>
<td>ball control</td>
</tr>
</tbody>
</table>
Layered Learning Paradigms

**DESCRIPTIONS:**

**Sequential Layered Learning:** Freeze parameters of layer after learning before learning of the next layer.

**Concurrent Layered Learning:** Keep parameters of layer open during learning of the next layer.
Layered Learning Paradigms

**PROBLEMS:**

**Sequential Layered Learning:** Can be too *limiting* in the joint layer policy search space

**Concurrent Layered Learning:** The increased dimensionality can make learning harder or intractible
Layered Learning Paradigms

SOLUTION:

**Overlapping Layered Learning:** Tradeoff between freezing or keeping open previous learned layers

Optimizes “seam” or overlap between behaviors: keeps some parts of previously learned layers open during subsequent learning
Overlapping Layered Learning

Sequential Layered Learning (SLL)

Concurrent Layered Learning (CLL)

Overlapping Layered Learning

Combining Independently Learned Behaviors (CILB)

Partial Concurrent Layered Learning (PCLL)

Previous Learned Layer Refinement (PLLKR)
Overlapping Layered Learning

Combining Independently Learned Behaviors (CILB)

Combining Independently Learned Behaviors: Behaviors learned independently and then combined by relearning subset of behaviors’ parameters

Sequential Layered Learning (SLL)

Concurrent Layered Learning (CLL)

Partial Concurrent Layered Learning (PCLL)

Previous Learned Layer Refinement (PLL)
Overlapping Layered Learning

Combining Independently Learned Behaviors: Behaviors learned independently and then combined by relearning subset of behaviors’ parameters

Partial Concurrent Layered Learning: Part, but not all, of a previously learned layer’s behaviors are left open
Overlapping Layered Learning

Sequential Layered Learning (SLL)

Concurrent Layered Learning (CLL)

Combining Independently Learned Behaviors (CILB)

Partial Concurrent Layered Learning (PCLL)

Previous Learned Layer Refinement (PLL)

Combining Independently Learned Behaviors: Behaviors learned independently and then combined by relearning subset of behaviors’ parameters

Partial Concurrent Layered Learning: Part, but not all, of a previously learned layer’s behaviors are left open

Previous Learned Layer Refinement: After a pair of layers is learned, part or all of the initial layer is unfrozen
RoboCup 3D Simulation Domain

- Teams of 11 vs 11 autonomous robots play soccer
- Realistic physics using Open Dynamics Engine (ODE)
- Simulated robots modeled after Aldebaron Nao robot
- Robot receives noisy visual information about environment
- Robots can communicate over limited bandwidth channel
RoboCup Champions 2011, 2012
RoboCup Champions 2011, 2012
Humanoid Walk Learning via Layered Learning and CMA-ES
• Parameterized double linear inverted pendulum model
RoboCup Champions 2011, 2012

Humanoid Walk Learning via Layered Learning and CMA-ES

- Parameterized **double linear inverted pendulum model**

CMA-ES
[Hansen, ’09]

- Stochastic, derivative-free, numerical optimization method
- Candidates sampled from **multidimensional Gaussian**
RoboCup Champions 2011, 2012

Humanoid Walk Learning via Layered Learning and CMA-ES

- Parameterized double linear inverted pendulum model

CMA-ES

- Stochastic, derivative-free, numerical optimization method
- Candidates sampled from multidimensional Gaussian
  - Mean maximizes likelihood of previous successes
  - Covariance update controls search step sizes

[Hansen, ’09]
RoboCup Champions 2011, 2012

Humanoid Walk Learning via Layered Learning and CMA-ES

- Parameterized double linear inverted pendulum model

CMA-ES
[Hansen, ’09]

- Stochastic, derivative-free, numerical optimization method
- Candidates sampled from multidimensional Gaussian
  - Mean maximizes likelihood of previous successes
  - Covariance update controls search step sizes

Initial walk
3 layers
No layered learning
Final walk
2 layers
Champs*2
19 learned behaviors for standing up, walking, and kicking

- CILB, PCLL, PLLR

Over 500 parameters optimized during the course of learning

- frozen, passed, seeded
Dribbling and Kicking the Ball in the Goal

- Four different walk parameter sets
  - Target/sprint/position + **approach ball to kick**
- Learn **fixed kick**
- Combine **kick with walk**: combine independent layers (CILB)
  - Overlap kick parameters for positioning
- Final **walk and kick**
**Scoring on a Kickoff**

- **Kickoffs indirect** (2 players must touch to score)
- Learn **fixed kick**
- Learn **touch** behavior interferes
- Combine **kick with touch**
  - Relearn position patterns: combine independent layers (CILB)
  - Learn new timing parameter: partial concurrent (PCLL)
Impact of Overlapping Layered Learning

1000 games vs. top 3 teams from 2013
## Impact of Overlapping Layered Learning

### 1000 games vs. top 3 teams from 2013

<table>
<thead>
<tr>
<th>Opponent</th>
<th>Full Team</th>
<th>No Kickoff</th>
<th>Dribble Only</th>
</tr>
</thead>
<tbody>
<tr>
<td>apollo3d</td>
<td>2.703 (0.041)</td>
<td>2.062 (0.038)</td>
<td>1.861 (0.034)</td>
</tr>
<tr>
<td>UTAustinVilla2013</td>
<td>1.589 (0.036)</td>
<td>1.225 (0.033)</td>
<td>0.849 (0.025)</td>
</tr>
<tr>
<td>fcportugal3d</td>
<td>3.991 (0.051)</td>
<td>3.189 (0.048)</td>
<td>1.584 (0.030)</td>
</tr>
</tbody>
</table>

**No Kickoff:** On kickoff, kick ball deep into opponent’s end

**Dribble Only:** No kicking
Repetition on Different Robot Types

Type 0: Standard Nao model
Type 1: Longer legs and arms
Type 2: Quicker moving legs
Type 3: Wider hips and longest legs and arms
Type 4: Added toes to foot
Repetition on Different Robot Types

Type 0: Standard Nao model
Type 1: Longer legs and arms
Type 2: Quicker moving legs
Type 3: Wider hips and longest legs and arms
Type 4: Added toes to foot

<table>
<thead>
<tr>
<th>Opponent</th>
<th>Type 0</th>
<th>Type 1</th>
<th>Type 2</th>
<th>Type 3</th>
<th>Type 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>apollo3d</td>
<td>1.787</td>
<td>1.819</td>
<td>1.820</td>
<td>1.543</td>
<td>2.827</td>
</tr>
<tr>
<td>UTAustinVilla2013</td>
<td>0.992</td>
<td>0.892</td>
<td>1.276</td>
<td>0.573</td>
<td>1.141</td>
</tr>
<tr>
<td>fcportugal3d</td>
<td>2.423</td>
<td>3.025</td>
<td>3.275</td>
<td>2.678</td>
<td>4.033</td>
</tr>
</tbody>
</table>
Repetition on Different Robot Types

Type 0: Standard Nao model
Type 1: Longer legs and arms
Type 2: Quicker moving legs
Type 3: Wider hips and longest legs and arms
Type 4: Added toes to foot

<table>
<thead>
<tr>
<th>Opponent</th>
<th>Type 0</th>
<th>Type 1</th>
<th>Type 2</th>
<th>Type 3</th>
<th>Type 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>apollo3d</td>
<td>1.787</td>
<td>1.819</td>
<td>1.820</td>
<td>1.543</td>
<td>2.827</td>
</tr>
<tr>
<td>UTAustinVilla2013</td>
<td>0.992</td>
<td>0.892</td>
<td>1.276</td>
<td>0.573</td>
<td>1.141</td>
</tr>
<tr>
<td>fcportugal3d</td>
<td>2.423</td>
<td>3.025</td>
<td>3.275</td>
<td>2.678</td>
<td>4.033</td>
</tr>
</tbody>
</table>

Computation per type

≈ 700k parameter sets evaluated
≈ 1.5 years compute time (≈ 50 hours on condor cluster)
RoboCup 2014

Won competition with *undefeated* record: outscored opps 52–0

<table>
<thead>
<tr>
<th>Opponent</th>
<th>Avg. Goal Diff.</th>
<th>Record (W-L-T)</th>
<th>Goals (F/A)</th>
<th>KO Score %</th>
</tr>
</thead>
<tbody>
<tr>
<td>BahiaRT</td>
<td>2.075 (0.030)</td>
<td>990-0-10</td>
<td>2092/17</td>
<td>96.2</td>
</tr>
<tr>
<td>FCPortugal</td>
<td>2.642 (0.034)</td>
<td>986-0-14</td>
<td>2748/106</td>
<td>83.4</td>
</tr>
<tr>
<td>magmaOffenburg</td>
<td>2.855 (0.035)</td>
<td>990-0-10</td>
<td>2864/9</td>
<td>88.3</td>
</tr>
<tr>
<td>RoboCanes</td>
<td>3.081 (0.046)</td>
<td>974-0-26</td>
<td>3155/74</td>
<td>69.4</td>
</tr>
<tr>
<td>FUT-K</td>
<td>3.236 (0.039)</td>
<td>998-0-2</td>
<td>3240/4</td>
<td>96.3</td>
</tr>
<tr>
<td>SEU_Jolly</td>
<td>4.031 (0.062)</td>
<td>995-0-5</td>
<td>4034/3</td>
<td>87.6</td>
</tr>
<tr>
<td>KarachiKoalas</td>
<td>5.681 (0.046)</td>
<td>1000-0-0</td>
<td>5682/1</td>
<td>87.5</td>
</tr>
<tr>
<td>ODENS</td>
<td>7.933 (0.041)</td>
<td>1000-0-0</td>
<td>7933/0</td>
<td>92.1</td>
</tr>
<tr>
<td>HfutEngine</td>
<td>8.510 (0.050)</td>
<td>1000-0-0</td>
<td>8510/0</td>
<td>94.7</td>
</tr>
<tr>
<td>Mithras3D</td>
<td>8.897 (0.041)</td>
<td>1000-0-0</td>
<td>8897/0</td>
<td>90.4</td>
</tr>
<tr>
<td>L3M-SIM</td>
<td>9.304 (0.043)</td>
<td>1000-0-0</td>
<td>9304/0</td>
<td>93.7</td>
</tr>
</tbody>
</table>
RoboCup 2014

Won competition with **undefeated** record: outscored opps 52–0

<table>
<thead>
<tr>
<th>Opponent</th>
<th>Avg. Goal Diff.</th>
<th>Record (W-L-T)</th>
<th>Goals (F/A)</th>
<th>KO Score %</th>
</tr>
</thead>
<tbody>
<tr>
<td>BahiaRT</td>
<td>2.075 (0.030)</td>
<td>990-0-10</td>
<td>2092/17</td>
<td>96.2</td>
</tr>
<tr>
<td>FCPortugal</td>
<td>2.642 (0.034)</td>
<td>986-0-14</td>
<td>2748/106</td>
<td>83.4</td>
</tr>
<tr>
<td>magmaOffenburg</td>
<td>2.855 (0.035)</td>
<td>990-0-10</td>
<td>2864/9</td>
<td>88.3</td>
</tr>
<tr>
<td>RoboCanes</td>
<td>3.081 (0.046)</td>
<td>974-0-26</td>
<td>3155/74</td>
<td>69.4</td>
</tr>
<tr>
<td>FUT-K</td>
<td>3.236 (0.039)</td>
<td>998-0-2</td>
<td>3240/4</td>
<td>96.3</td>
</tr>
<tr>
<td>SEU_Jolly</td>
<td>4.031 (0.062)</td>
<td>995-0-5</td>
<td>4034/3</td>
<td>87.6</td>
</tr>
<tr>
<td>KarachiKoalas</td>
<td>5.681 (0.046)</td>
<td>1000-0-0</td>
<td>5682/1</td>
<td>87.5</td>
</tr>
<tr>
<td>ODENS</td>
<td>7.933 (0.041)</td>
<td>1000-0-0</td>
<td>7933/0</td>
<td>92.1</td>
</tr>
<tr>
<td>HfutEngine</td>
<td>8.510 (0.050)</td>
<td>1000-0-0</td>
<td>8510/0</td>
<td>94.7</td>
</tr>
<tr>
<td>Mithras3D</td>
<td>8.897 (0.041)</td>
<td>1000-0-0</td>
<td>8897/0</td>
<td>90.4</td>
</tr>
<tr>
<td>L3M-SIM</td>
<td>9.304 (0.043)</td>
<td>1000-0-0</td>
<td>9304/0</td>
<td>93.7</td>
</tr>
</tbody>
</table>

- **After: 11,000 games: won all by 67 (no losses)**
RoboCup 2014

Won competition with **undefeated** record: outscored opps 52–0

<table>
<thead>
<tr>
<th>Opponent</th>
<th>Avg. Goal Diff.</th>
<th>Record (W-L-T)</th>
<th>Goals (F/A)</th>
<th>KO Score %</th>
</tr>
</thead>
<tbody>
<tr>
<td>BahiaRT</td>
<td>2.075 (0.030)</td>
<td>990-0-10</td>
<td>2092/17</td>
<td>96.2</td>
</tr>
<tr>
<td>FCPortugal</td>
<td>2.642 (0.034)</td>
<td>986-0-14</td>
<td>2748/106</td>
<td>83.4</td>
</tr>
<tr>
<td>magmaOffenburg</td>
<td>2.855 (0.035)</td>
<td>990-0-10</td>
<td>2864/9</td>
<td>88.3</td>
</tr>
<tr>
<td>RoBoCanes</td>
<td>3.081 (0.046)</td>
<td>974-0-26</td>
<td>3155/74</td>
<td>69.4</td>
</tr>
<tr>
<td>FUT-K</td>
<td>3.236 (0.039)</td>
<td>998-0-2</td>
<td>3240/4</td>
<td>96.3</td>
</tr>
<tr>
<td>SEU_Jolly</td>
<td>4.031 (0.062)</td>
<td>995-0-5</td>
<td>4034/3</td>
<td>87.6</td>
</tr>
<tr>
<td>KarachiKoalas</td>
<td>5.681 (0.046)</td>
<td>1000-0-0</td>
<td>5682/1</td>
<td>87.5</td>
</tr>
<tr>
<td>ODENS</td>
<td>7.933 (0.041)</td>
<td>1000-0-0</td>
<td>7933/0</td>
<td>92.1</td>
</tr>
<tr>
<td>HfutEngine</td>
<td>8.510 (0.050)</td>
<td>1000-0-0</td>
<td>8510/0</td>
<td>94.7</td>
</tr>
<tr>
<td>Mithras3D</td>
<td>8.897 (0.041)</td>
<td>1000-0-0</td>
<td>8897/0</td>
<td>90.4</td>
</tr>
<tr>
<td>L3M-SIM</td>
<td>9.304 (0.043)</td>
<td>1000-0-0</td>
<td>9304/0</td>
<td>93.7</td>
</tr>
</tbody>
</table>

- After: **11,000 games**: won all by 67 (**no losses**)
- **Highlights** from Final vs. RoBoCanes (University of Miami)
RoboCup 2014

Won competition with *undefeated* record: outscored opps 52–0

<table>
<thead>
<tr>
<th>Opponent</th>
<th>Avg. Goal Diff.</th>
<th>Record (W-L-T)</th>
<th>Goals (F/A)</th>
<th>KO Score %</th>
</tr>
</thead>
<tbody>
<tr>
<td>BahiaRT</td>
<td>2.075 (0.030)</td>
<td>990-0-10</td>
<td>2092/17</td>
<td>96.2</td>
</tr>
<tr>
<td>FCPortugal</td>
<td>2.642 (0.034)</td>
<td>986-0-14</td>
<td>2748/106</td>
<td>83.4</td>
</tr>
<tr>
<td>magmaOffenburg</td>
<td>2.855 (0.035)</td>
<td>990-0-10</td>
<td>2864/9</td>
<td>88.3</td>
</tr>
<tr>
<td>RoboCanes</td>
<td>3.081 (0.046)</td>
<td>974-0-26</td>
<td>3155/74</td>
<td>69.4</td>
</tr>
<tr>
<td>FUT-K</td>
<td>3.236 (0.039)</td>
<td>998-0-2</td>
<td>3240/4</td>
<td>96.3</td>
</tr>
<tr>
<td>SEU_Jolly</td>
<td>4.031 (0.062)</td>
<td>995-0-5</td>
<td>4034/3</td>
<td>87.6</td>
</tr>
<tr>
<td>KarachiKoalas</td>
<td>5.681 (0.046)</td>
<td>1000-0-0</td>
<td>5682/1</td>
<td>87.5</td>
</tr>
<tr>
<td>ODENS</td>
<td>7.933 (0.041)</td>
<td>1000-0-0</td>
<td>7933/0</td>
<td>92.1</td>
</tr>
<tr>
<td>HfutEngine</td>
<td>8.510 (0.050)</td>
<td>1000-0-0</td>
<td>8510/0</td>
<td>94.7</td>
</tr>
<tr>
<td>Mithras3D</td>
<td>8.897 (0.041)</td>
<td>1000-0-0</td>
<td>8897/0</td>
<td>90.4</td>
</tr>
<tr>
<td>L3M-SIM</td>
<td>9.304 (0.043)</td>
<td>1000-0-0</td>
<td>9304/0</td>
<td>93.7</td>
</tr>
</tbody>
</table>

- **After:** *11,000 games*: won all by 67 (*no losses*)
- **Highlights** from Final vs. RoboCanes (University of Miami)
- Successfully *defended championship* in 2015 (87–1)
RoboCup 2014

Won competition with *undefeated* record: outscored opps 52–0

<table>
<thead>
<tr>
<th>Opponent</th>
<th>Avg. Goal Diff.</th>
<th>Record (W-L-T)</th>
<th>Goals (F/A)</th>
<th>KO Score %</th>
</tr>
</thead>
<tbody>
<tr>
<td>BahiaRT</td>
<td>2.075 (0.030)</td>
<td>990-0-10</td>
<td>2092/17</td>
<td>96.2</td>
</tr>
<tr>
<td>FCPortugal</td>
<td>2.642 (0.034)</td>
<td>986-0-14</td>
<td>2748/106</td>
<td>83.4</td>
</tr>
<tr>
<td>magmaOffenburg</td>
<td>2.855 (0.035)</td>
<td>990-0-10</td>
<td>2864/9</td>
<td>88.3</td>
</tr>
<tr>
<td>RoboCanes</td>
<td>3.081 (0.046)</td>
<td>974-0-26</td>
<td>3155/74</td>
<td>69.4</td>
</tr>
<tr>
<td>FUT-K</td>
<td>3.236 (0.039)</td>
<td>998-0-2</td>
<td>3240/4</td>
<td>96.3</td>
</tr>
<tr>
<td>SEU_Jolly</td>
<td>4.031 (0.062)</td>
<td>995-0-5</td>
<td>4034/3</td>
<td>87.6</td>
</tr>
<tr>
<td>KarachiKoalas</td>
<td>5.681 (0.046)</td>
<td>1000-0-0</td>
<td>5682/1</td>
<td>87.5</td>
</tr>
<tr>
<td>ODENS</td>
<td>7.933 (0.041)</td>
<td>1000-0-0</td>
<td>7933/0</td>
<td>92.1</td>
</tr>
<tr>
<td>HfutEngine</td>
<td>8.510 (0.050)</td>
<td>1000-0-0</td>
<td>8510/0</td>
<td>94.7</td>
</tr>
<tr>
<td>Mithras3D</td>
<td>8.897 (0.041)</td>
<td>1000-0-0</td>
<td>8897/0</td>
<td>90.4</td>
</tr>
<tr>
<td>L3M-SIM</td>
<td>9.304 (0.043)</td>
<td>1000-0-0</td>
<td>9304/0</td>
<td>93.7</td>
</tr>
</tbody>
</table>

- **After:** 11,000 games: won all by 67 (*no losses*)
- **Highlights** from Final vs. RoboCanes (University of Miami)
- Successfully **defended championship** in 2015 (87–1)
- **RoboCup 2016** in June: Leipzig, Germany
RoboCup 2014

Won competition with **undefeated** record: outscored opps 52–0

<table>
<thead>
<tr>
<th>Opponent</th>
<th>Avg. Goal Diff.</th>
<th>Record (W-L-T)</th>
<th>Goals (F/A)</th>
<th>KO Score %</th>
</tr>
</thead>
<tbody>
<tr>
<td>BahiaRT</td>
<td>2.075 (0.030)</td>
<td>990-0-10</td>
<td>2092/17</td>
<td>96.2</td>
</tr>
<tr>
<td>FCPortugal</td>
<td>2.642 (0.034)</td>
<td>986-0-14</td>
<td>2748/106</td>
<td>83.4</td>
</tr>
<tr>
<td>magmaOffenburg</td>
<td>2.855 (0.035)</td>
<td>990-0-10</td>
<td>2864/9</td>
<td>88.3</td>
</tr>
<tr>
<td>RoboCanes</td>
<td>3.081 (0.046)</td>
<td>974-0-26</td>
<td>3155/74</td>
<td>69.4</td>
</tr>
<tr>
<td>FUT-K</td>
<td>3.236 (0.039)</td>
<td>998-0-2</td>
<td>3240/4</td>
<td>96.3</td>
</tr>
<tr>
<td>SEU_Jolly</td>
<td>4.031 (0.062)</td>
<td>995-0-5</td>
<td>4034/3</td>
<td>87.6</td>
</tr>
<tr>
<td>KarachiKoalas</td>
<td>5.681 (0.046)</td>
<td>1000-0-0</td>
<td>5682/1</td>
<td>87.5</td>
</tr>
<tr>
<td>ODENS</td>
<td>7.933 (0.041)</td>
<td>1000-0-0</td>
<td>7933/0</td>
<td>92.1</td>
</tr>
<tr>
<td>HfutEngine</td>
<td>8.510 (0.050)</td>
<td>1000-0-0</td>
<td>8510/0</td>
<td>94.7</td>
</tr>
<tr>
<td>Mithras3D</td>
<td>8.897 (0.041)</td>
<td>1000-0-0</td>
<td>8897/0</td>
<td>90.4</td>
</tr>
<tr>
<td>L3M-SIM</td>
<td>9.304 (0.043)</td>
<td>1000-0-0</td>
<td>9304/0</td>
<td>93.7</td>
</tr>
</tbody>
</table>

- **After:** 11,000 games: won all by 67 (no losses)
- **Highlights** from Final vs. RoboCanes (University of Miami)
- Successfully **defended championship** in 2015 (87–1)
- **RoboCup 2016** in June: Leipzig, Germany
- More info: [www.cs.utexas.edu/~AustinVilla/sim/3dsimulation/](http://www.cs.utexas.edu/~AustinVilla/sim/3dsimulation/)
Autonomous Learning Agents

- Learning in practice: Layered learning
- Learning in MAS: Ad hoc teamwork
Teamwork
Teamwork
Teamwork

Typical scenario: **pre-coordination**
- People practice together
- Robots given **coordination languages, protocols**
- “Locker room agreement” [Stone & Veloso, ’99]
Ad Hoc Teams

- Ad hoc team player is an individual
  - Unknown teammates (*programmed by others*)
Ad Hoc Teams

- Ad hoc team player is an individual
  - Unknown teammates (programmed by others)

- May or may not be able to communicate
Ad Hoc Teams

- Ad hoc team player is an individual
  - Unknown teammates *(programmed by others)*
- May or **may not** be able to communicate
- Teammates likely *sub-optimal*: no control
Ad Hoc Teams

- Ad hoc team player is an individual
  - Unknown teammates (programmed by others)

- May or may not be able to communicate

- Teammates likely sub-optimal: no control
Ad Hoc Teams

- Ad hoc team player is an **individual**
  - Unknown teammates (**programmed by others**)

- May or **may not** be able to communicate

- Teammates likely **sub-optimal**: no control

**Challenge:** Create a good team player
Create an autonomous agent that is able to efficiently and robustly collaborate with previously unknown teammates on tasks to which they are all individually capable of contributing as team members.
Create an autonomous agent that is able to efficiently and robustly collaborate with previously unknown teammates on tasks to which they are all individually capable of contributing as team members.

- Aspects can be approached theoretically
Ad Hoc Teamwork Challenge [AAAI’10]

Create an autonomous agent that is able to efficiently and robustly collaborate with previously unknown teammates on tasks to which they are all individually capable of contributing as team members.

- Aspects can be approached theoretically
- Ultimately an empirical challenge
Technical Requirements

- Assess capabilities of other agents (teammate modeling)
Technical Requirements

- Assess capabilities of other agents (teammate modeling)
- Assess the other agents’ knowledge states
Technical Requirements

- Assess capabilities of other agents *(teammate modeling)*
- Assess the other agents’ knowledge states
- Estimate **effects** of actions on teammates
Technical Requirements

- Assess capabilities of other agents (teammate modeling)
- Assess the other agents’ knowledge states
- Estimate effects of actions on teammates
- Be prepared to interact with many types of teammates:
  - May or may not be able to communicate
  - May be more or less mobile
  - May be better or worse at sensing
Technical Requirements

- Assess capabilities of other agents (teammate modeling)
- Assess the other agents’ knowledge states
- Estimate effects of actions on teammates
- Be prepared to interact with many types of teammates:
  - May or may not be able to communicate
  - May be more or less mobile
  - May be better or worse at sensing

A good team player’s best actions will differ depending on its teammates’ characteristics.
Initial Theoretical Progress

- Ultimately an empirical challenge
- Aspects can be approached *theoretically*
Initial Theoretical Progress

- Ultimately an empirical challenge
- **Aspects can be approached theoretically**

Be **prepared** to interact with many types of teammates
Initial Theoretical Progress

- Ultimately an empirical challenge
- **Aspects can be approached theoretically**

Be **prepared** to interact with many types of teammates

- Minimal representative scenarios
  - One teammate, no communication
  - Fixed and known behavior
Scenarios

- Cooperative iterated normal form game
  [w/ Kaminka, Rosenschein, and Agmon—AIJ’13]

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$M_1$</td>
<td>$b_0$</td>
<td>$b_1$</td>
</tr>
<tr>
<td>$a_0$</td>
<td>25</td>
<td>1</td>
</tr>
<tr>
<td>$a_1$</td>
<td>10</td>
<td>30</td>
</tr>
<tr>
<td>$a_2$</td>
<td>0</td>
<td>33</td>
</tr>
</tbody>
</table>

- Cooperative $k$-armed bandit
  [w/ Kraus—AAMAS’10]
Team Knowledge: Does the ad hoc agent know what its teammates’ actions will be for a given state, before interacting with them?
Team Knowledge: Does the ad hoc agent know what its teammates’ actions will be for a given state, before interacting with them?

Environment Knowledge: Does the ad hoc agent know the transition and reward distribution given a joint action and state before interacting with the environment?
Dimensions [Barrett & Stone, AAMAS’12]

**Team Knowledge:** Does the ad hoc agent know what its teammates’ actions will be for a given state, before interacting with them?

**Environment Knowledge:** Does the ad hoc agent know the transition and reward distribution given a joint action and state before interacting with the environment?

**Reactivity of teammates:** How much does the ad hoc agent’s actions affect those of its teammates?
Autonomous Learning Agents

- Learning in practice: Layered learning
- Learning in MAS: Ad hoc teamwork
  - Evaluation framework
Autonomous Learning Agents

- Learning in practice: Layered learning
- Learning in MAS: Ad hoc teamwork
  - Evaluation framework
  - Planning and Learning to Adapt Swiftly to Teammates to Improve Cooperation (PLASTIC) [Barrett,’14]
Learning in practice: Layered learning

**Learning in MAS: Ad hoc teamwork**
- Evaluation framework
- Planning and Learning to Adapt Swiftly to Teammates to Improve Cooperation (PLASTIC) [Barrett,’14]
- Other ad hoc teamwork research
Empirical Evaluation
Evaluation: A Metric
Evaluation: A Metric

Most meaningful when $a_0$ and $a_1$ have similar individual competencies
Evaluation: Domain Consisting of Tasks
Evaluation: Set of Possible Teammates
Evaluation: Draw a Random Task

\[ a_0 \quad a_1 \]

D

A
Evaluation: Random Team, Check Comp
Evaluation: Replace Random with a0
Evaluation: Then a1 — Evaluate Diff
Evaluation: Repeat
**Evaluate**($a_0$, $a_1$, $A$, $D$)

- Initialize performance (reward) counters $r_0$ and $r_1$ for agents $a_0$ and $a_1$ respectively to $r_0 = r_1 = 0$.

- Repeat:
  - Sample a task $d$ from $D$.
  - Randomly draw a subset of agents $B$, $|B| \geq 2$, from $A$ such that $E[s(B, d)] \geq s_{min}$.
  - Randomly select one agent $b \in B$ to remove from the team to create the team $B^-$.
  - Increment $r_0$ by $s(\{a_0\} \cup B^-, d)$
  - Increment $r_1$ by $s(\{a_1\} \cup B^-, d)$

- If $r_0 > r_1$ then we conclude that $a_0$ is a better ad-hoc team player than $a_1$ in domain $D$ over the set of possible teammates $A$. 
Learning in practice: Layered learning

Learning in MAS: Ad hoc teamwork
  - Evaluation framework
  - Planning and Learning to Adapt Swiftly to Teammates to Improve Cooperation (PLASTIC) [Barrett, ’14]
  - Other ad hoc teamwork research
Single Agent Learning
Multiagent Coordination
Ad Hoc Teamwork
PLASTIC Overview

- Planning and Learning to Adapt Swiftly to Teammates to Improve Cooperation (PLASTIC)
PLASTIC Overview

- Planning and Learning to Adapt Swiftly to Teammates to Improve Cooperation (PLASTIC)
- Learn about previous teammates
Planning and Learning to Adapt Swiftly to Teammates to Improve Cooperation (PLASTIC)

Learn about previous teammates

Reuse this knowledge with new teammates
PLASTIC Overview

- Planning and Learning to Adapt Swiftly to Teammates to Improve Cooperation (PLASTIC)
- Learn about previous teammates
- Reuse this knowledge with new teammates
- Determine which previous teammates are most similar to the new ones
PLASTIC Overview

- Planning and Learning to Adapt Swiftly to Teammates to Improve Cooperation (PLASTIC)
- Learn about previous teammates
- Reuse this knowledge with new teammates
- Determine which previous teammates are most similar to the new ones
- Two variants: PLASTIC–Model and PLASTIC–Policy
Overview of PLASTIC

- Learned Knowledge of Teammates
- Hand Coded Prior Knowledge

Ad Hoc Agent

Inference of Teammate Type

Updates

Teammate Knowledge

Action Selection

State, Reward

Environment

Team

Agent

Agent
Overview of PLASTIC
PLASTIC: Expert Knowledge

- Allow experts to provide prior knowledge
- Information about teammate behaviors or how to adapt to teammates
- Prior belief distribution over teammate behaviors
Agent has extensive interactions with previous teammates

Learn about previous teammates

Use this knowledge to cooperate with new teammates
PLASTIC: Inferring Teammate Type

- Observe the actions of the teammates
- Determine the probability of a known teammate type taking the observed actions
- Update the distribution over the teammate types using a bounded loss version of Bayes’ rule
PLASTIC: Action Selection

- Given the distribution over teammate types
- Given current world state
- Determine best action to take
PLASTIC–Model Motivation

- Model-based approach
- Adapts quickly to new teammates
- Reuses models of past teammates
PLASTIC–Model Motivation

- Model-based approach
- Adapts quickly to new teammates
- Reuses models of past teammates
- Given the true model of the environment and teammates, can calculate the optimal policy
- Can select actions given a distribution over model types
Overview of PLASTIC–Model
Overview of PLASTIC–Model

[Diagram of the PLASTIC–Model framework, showing components such as Learned Teammate Modes, Hand Coded Teammate Models, Ad Hoc Agent, Model Selection, Teammate Models, Planning, Team, Environment, State Reward, Joint Action, and Teammates' Actions.]
PLASTIC—Model: Expert Knowledge

- Model-based approach
- Expert provides teammate models
- Hand-coded behaviors of potential teammates
PLASTIC–Model: Learn about Previous Teammates

- Collect samples of past teammates
- Mapping from states to actions
PLASTIC–Model: Learn about Previous Teammates

- Collect samples of past teammates
- Mapping from states to actions
- Supervised learning problem
- Use existing learning algorithms, such as decision trees
PLASTIC–Model: Learn about Previous Teammates

- Collect samples of past teammates
- Mapping from states to actions
- Supervised learning problem
- Use existing learning algorithms, such as decision trees
- Can use transfer learning, such as TwoStageTransfer
PLASTIC–Model: Inferring Teammate Type

- Update models using observed actions
- Use Bayes’ rule
PLASTIC–Model: Inferring Teammate Type

- Update models using observed actions
- Use Bayes’ rule
- But may have some bad predictions
- Use bounded loss
PLASTIC–Model: Inferring Teammate Type

- Update models using observed actions
- Use Bayes’ rule
- But may have some bad predictions
- Use bounded loss

$$\text{loss} = 1 - P(\text{actions} | \text{model})$$

$$P(\text{model} | \text{actions}) \propto (1 - \eta \times \text{loss}) \times P(\text{model})$$
PLASTIC–Model: Action Selection

- Select best action
- Know model
- Know distribution over teammates
PLASTIC–Model: Action Selection

- Select best action
- Know model
- Know distribution over teammates
- Solve using MDP planners, such as UCT
PLASTIC–Policy Motivation

- Policy-based approach
- Reuses policies for cooperating with past teammates
- Adapts quickly to new teammates
PLASTIC–Policy Motivation

- Policy-based approach
- Reuses policies for cooperating with past teammates
- Adapts quickly to new teammates
- Policy-based methods better handle complex, noisy domains
  - Many robotic tasks have been better solved using policy-based approaches
- Fast online computation
Overview of PLASTIC–Policy
Overview of PLASTIC–Policy

Ad Hoc Agent

Policy Selection

Policies

Action Selection

Environment

State

Reward

Team

Agent

Hand Coded Policies

Learned Policies for Teammates

Updates

Tried by

Ad Hoc Agent’s Action

Teammates’ Actions

Joint Action

Peter Stone (UT Austin)  Autonomous Learning Agents  57
Policy-based approach

Expert provides policies for cooperating with teammates

Hand-coded policies for behaving intelligently
PLASTIC–Policy: Learn about Previous Teammates

- Collect samples of past teammates

\[ \langle s, a, r, s' \rangle \]
Collect samples of past teammates

\( \langle s, a, r, s' \rangle \)

Use existing policy learning algorithms, such as fitted Q iteration
PLASTIC–Policy: Inferring Teammate Type

- As in PLASTIC–Model
- Update models using observed actions
- Use Bayes’ rule with bounded loss
- But do not have full model
As in PLASTIC–Model

- Update models using observed actions
- Use Bayes’ rule with bounded loss
- But do not have full model
- Estimate using a nearest neighbors transition function
PLASTIC–Policy: Action Selection

- Straightforward

- Use policy with highest probability

- Select best action for policy
Test the hypothesis that PLASTIC enables agents to quickly adapt to new teammates in a variety of possible ad hoc teamwork scenarios.
Overview of Empirical Results

- Test the hypothesis that PLASTIC enables agents to quickly adapt to new teammates in a variety of possible ad hoc teamwork scenarios
- Test in 3 domains: Bandit, Pursuit, and HFO (simulated soccer)
Pursuit Domain

- Grid world - Torus
- 4 Predators and 1 Prey
- Goal is to capture the prey as quickly as possible
- Fully observable
- 5 actions: Stay still, up, down, left, and right
Pursuit Domain

- Grid world - Torus
- 4 Predators and 1 Prey
- Goal is to capture the prey as quickly as possible
- Fully observable
- 5 actions: Stay still, up, down, left, and right
Experimental Methodology

- Agent replaces single teammate in otherwise coherent team
- Adapts based on knowledge learned from previous teammates
Motivation

Do we need to do different things with different teammates?
Do we need to do different things with different teammates? YES!

![Bar chart showing comparison of times prey captured between Plan(Correct) and Plan(Incorrect) for different teammate types: GR and TA.](chart.png)
Cooperating with Unknown Teammates

- **Baselines:**
  - Plan with True model
  - Match teammates behavior
- Learn models of past teammates
- Select models based on probability of observed actions
- Plan using UCT - a Monte Carlo tree search

![Graph showing comparison between different planning methods.](image_url)

[Barrett & Stone, AAMAS’11]
Partially Observed Teammates

- Few observations of current teammates
- Many observations of past teammates
- Use transfer learning to improve models
- TwoStageTransfer - efficiently combines information coming from many sources

[Barrett & Stone, AAAI’13]
Testbed Domains

- Agent replaces single teammate in otherwise coherent team
- Adapts based on knowledge learned from previous teammates
Learning in practice: Layered learning

**Learning in MAS: Ad hoc teamwork**
- Evaluation framework
- Planning and Learning to Adapt Swiftly to Teammates to Improve Cooperation (PLASTIC) [Barrett,’14]
- Other ad hoc teamwork research
Synergy Graphs [Liemhetcharat & Veloso, ’13]

- Given a task and potential teammates
Synergy Graphs [Liemhetcharat & Veloso, ’13]

- Given a task and potential teammates
- Select a team that works well together
Representation, Planning and Learning of Dynamic Ad Hoc Robot Teams

Somchaya Liemhetcharat, 2013
Synergy Graph Model

- We contribute the Synergy Graph model, where:
  - Every **agent** is represented with a **vertex**
    - We will select a subset of the agents for the team
  - Each vertex has a **Normal distribution** that represents the agent’s **capability**
    - Represents the agent’s non-deterministic capability in a dynamic world
  - Edges represent the **task-based relationship** among the agents
    - Distance between agents in the graph correspond to how well they work together

Suppose there is a task to rescue an injured person:
- \(a_s\) (stretcher), \(a_c\) (CPR) and \(a_d\) (driver) are agents in an ambulance
- \(h_s\) (surgeon) and \(h_i\) (intern) are agents in a hospital

\[
C_{aa} \sim \mathcal{N}(5.2, 1.3) \quad C_{ac} \sim \mathcal{N}(17.2, 10.9)
\]

Team of agents in the **ambulance**:
\[
S(\{a_s, a_d, a_c\}) \sim \mathcal{N}(17.8, 2.8)
\]

Replacing the CPR agent with the surgeon:
\[
S(\{a_s, a_d, h_s\}) \sim \mathcal{N}(12.1, 0.3)
\]

Adding the surgeon:
\[
S(\{a_s, a_d, a_c, h_s\}) \sim \mathcal{N}(17.8, 0.8)
\]

Team of all the agents:
\[
S(\{a_s, a_d, a_c, h_s, h_i\}) \sim \mathcal{N}(13.0, 0.3)
\]
Type-Based Method [Albrecht & Ramamoorthy, ’16]

Type-based method:
- Hypothesize types (behaviors) for agents, e.g. from data
- Compute beliefs over types based on observed actions
- Plan own actions with respect to most likely types

Useful for ad hoc teamwork:
- **Flexible**: can hypothesize any behavior as type
- **Efficient**: fast learning, plan in unknown state space

Tutorial at AAAI-16:
“Type-based Methods for Interaction in Multiagent Systems”
S. Albrecht, P. Doshi
Type-Based Method

History

Agent

Belief

Planning

Own Action

Type 1

Type 2

Type 3

Action

Action

Action
Type-Based Method

Model & Algorithm:
- Stochastic Bayesian Game
- **Harsanyi-Bellman Ad Hoc Coordination** (HBA)
  → combines Bayes-Nash equilibrium and Bellman optimality

Questions:
- How to compute posterior beliefs and when correct?
- What long-term impact do prior beliefs have?
- When are we “optimal”, even if types incorrect?
- How to decide if hypothesized types incorrect?

S. Albrecht, S. Ramamoorthy
Belief and Truth in Hypothesised Behaviours
Flocking + Ad Hoc Teamwork

- Flocking animals follow a simple local behavior rule
  - Update behavior based on behavior of surrounding animals
- Resulting group behavior is cohesive
Flocking + Ad Hoc Teamwork

- Flocking animals follow a simple local behavior rule
  - Update behavior based on behavior of surrounding animals
- Resulting group behavior is cohesive

How should ad hoc agents behave so as to influence the rest of the flock to adopt a desired behavior as quickly as possible?
Flocking + Ad Hoc Teamwork

Influencing the flock to orient towards a particular orientation

Influencing the flock to travel around a particular area
Ad hoc teamwork at RoboCup [Genter et. al, ’15]

- Drop-in player challenges in 3 leagues
Ad hoc teamwork at RoboCup [Genter et. al, ’15]

- Drop-in player challenges in 3 leagues
Related Work

Multiagent learning  [Claus & Boutilier, '98],[Littman, '01],[Conitzer & Sandholm, '03],
[Powers & Shoham, '05],[Chakraborty & Stone, '08]

Opponent Modeling
- Intended plan recognition  [Sidner, '85], [Lochbaum,'91],
  [Carberry, '01]
- SharedPlans
- Recursive Modeling

Human-Robot-Agent Teams
- Overlapping but different challenges, including HRI  
  [Klein, '04]
- Out of scope

Much More  pertaining to specific teammate characteristics
Ad Hoc Teams

- Ad hoc team player is an **individual**
  - Unknown teammates *(programmed by others)*
- Teammates likely **sub-optimal**: no control

**Challenge:** Create a good team player
Ad Hoc Teams

- Ad hoc team player is an individual
  - Unknown teammates (programmed by others)
- Teammates likely sub-optimal: no control

**Challenge:** Create a good team player

- Introduced as AAAI Challenge Problem

[AAAI’10]
Ad Hoc Teams

- Ad hoc team player is an individual
  - Unknown teammates (programmed by others)
- Teammates likely sub-optimal: no control

**Challenge:** Create a good team player

- Introduced as AAAI Challenge Problem
  - Theory: repeated games, bandits

[AAAI’10]
[AIJ’13]
Ad Hoc Teams

- Ad hoc team player is an individual
  - Unknown teammates (programmed by others)
- Teammates likely sub-optimal: no control

**Challenge:** Create a good team player

- Introduced as AAAI Challenge Problem [AAAI’10]
  - Theory: repeated games, bandits [AIJ’13]
  - Experiments: pursuit, flocking, ... [Barrett, Genter, & Stone, ’12]
Ad Hoc Teams

- Ad hoc team player is an individual
  - Unknown teammates (programmed by others)
- Teammates likely sub-optimal: no control

**Challenge:** Create a good team player

- Introduced as AAAI Challenge Problem [AAAI’10]
  - Theory: repeated games, bandits [AIJ’13]
  - Experiments: pursuit, flocking, . . . [Barrett, Genter, & Stone, ’12]
  - Community: MIPC Workshops, JAAMAS issue, & . . .
Ad Hoc Teams

- Ad hoc team player is an individual
  - Unknown teammates (programmed by others)
- Teammates likely sub-optimal: no control

**Challenge:** Create a good team player

- Introduced as AAAI Challenge Problem
  - Theory: repeated games, bandits [AIJ’13]
  - Experiments: pursuit, flocking, . . . [Barrett, Genter, & Stone, ’12]
  - Community: MIPC Workshops, JAAMAS issue, & . . . you!
Autonomous Learning Agents

- Learning in practice: Layered learning
- Learning in MAS: Ad hoc teamwork
Research Question

To what degree can autonomous intelligent agents learn in the presence of teammates and/or adversaries in real-time, dynamic domains?

Research Areas

- Autonomous agents
- Machine learning
  - Reinforcement learning
- Multiagent systems
  - Trading agents
- Robotics
Selected RL Contributions

- Human interaction
Selected RL Contributions

- Human interaction
  - Advice, Demonstration
Selected RL Contributions

- Human interaction
  - Advice, Demonstration
  - Positive/Negative Feedback

[Knox & Stone, ’09]
Selected RL Contributions

- Human interaction
  - Advice, Demonstration
  - Positive/Negative Feedback

- Transfer learning for RL

[Knox & Stone, '09]
[Taylor & Stone, '07]
Selected RL Contributions

- Human interaction
  - Advice, Demonstration
  - Positive/Negative Feedback
- Transfer learning for RL
- TEXPLORE for Robot RL

[Knox & Stone, ’09]
[Taylor & Stone, ’07]
[Hester & Stone, ’13]
Selected RL Contributions

- **Human interaction**
  - Advice, Demonstration
  - Positive/Negative Feedback

- **Transfer** learning for RL

- **TEXPLORE** for Robot RL
  - Sample efficient; real-time
  - Continuous state; delayed effects

[Knox & Stone, ’09]
[Taylor & Stone, ’07]
[Hester & Stone, ’13]
Selected RL Contributions

- Human interaction
  - Advice, Demonstration
  - Positive/Negative Feedback

- Transfer learning for RL

- TEXPLORE for Robot RL
  - Sample efficient; real-time
  - Continuous state; delayed effects

- Deep RL in continuous action spaces

[Knox & Stone, ’09]
[Taylor & Stone, ’07]
[Hester & Stone, ’13]
[Hausknecht & Stone, ’16]
Selected RL Contributions

- Human interaction
  - Advice, Demonstration
  - Positive/Negative Feedback

- Transfer learning for RL

- TEXPLORE for Robot RL
  - Sample efficient; real-time
  - Continuous state; delayed effects

- Deep RL in continuous action spaces

- Curriculum Learning – Narvekar: Learning IV, Waterfront 2
Power TAC

- National grid
- Renewable production
- Commercial/residential consumers
- Electricity generation companies
- Wholesale Market
- Balancing Market
- Tariff Market
- Competing broker agents
Acknowledgements

Patrick MacAlpine  Shimon Whiteson  Sam Barrett  Katie Genter  Noa Agmon
Acknowledgements

Patrick MacAlpine
Shimon Whiteson
Sam Barrett
Katie Genter
Noa Agmon

Brad Knox
Matt Taylor
Todd Hester
Matthew Hausknecht
Sanmit Narvekar
Daniel Urieli
Acknowledgements

Patrick MacAlpine
Shimon Whiteson
Sam Barrett
Katie Genter
Noa Agmon

Brad Knox
Matt Taylor
Todd Hester
Matthew Hausknecht
Sanmit Narvekar
Daniel Urieli

Piyush Khandelwal
Jake Menashe
Elad Liebman
Josiah Hanna
Acknowledgements

Patrick MacAlpine
Shimon Whiteson
Sam Barrett
Katie Genter
Noa Agmon
Brad Knox
Matt Taylor
Todd Hester
Matthew Hausknecht
Sanmit Narvekar
Daniel Urieli
Piyush Khandelwal
Jake Menashe
Elad Liebman
Josiah Hanna

LARG (past and present), NSF, ONR, AFOSR, AFRL
Layered Learning and Ad Hoc Teamwork

To what degree can autonomous intelligent agents learn in the presence of teammates and/or adversaries in real-time, dynamic domains?