Multi-Robot Planning, Execution, and Some Learning

Manuela M. Veloso
Computer Science Department
Carnegie Mellon University

Simmons and Veloso, PEL 2010

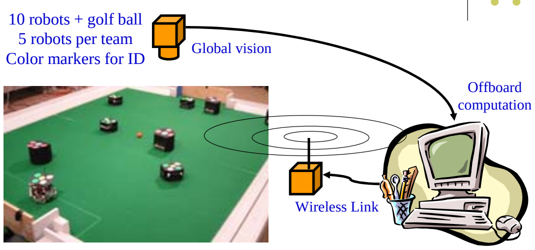
Multi-Robot – Robot Soccer

CMUnited
CMDragons
wheeled, offboard perception & control

CMTrio,
CMPack,
CMDash
AIBOs, 4-legged, fully autonomous

CMBalance:
Segway soccer
human-robot teams
(with B. Browning et al.)

"Small-Size" Robot Soccer



Complete system is autonomous as a whole

STP Behavioral Architecture

- **Skills**
  - Actions
  - Low-level control scripts to achieve *plain* goals
- **Tactics**
  - Sequences of actions
  - Combinations of skills to achieve *task* goals
- **Plays**
  - Web of tactics
  - Plans of tactics to achieve *team* goals

Team Strategies – Plays
Applicability, Termination, Roles

PLAY Two Attackers, Corner Dribble 1

APPLICABLE offense in_their_corner
DONE aborted !offense
TIMEOUT 15

ROLE 1
dribble_to_shoot { R { B 1100 800 } { B 700 800 } 300}
shoot A
none
ROLE 2
block 320 900 - 1
ROLE 3
position_for_pass { R { B 1000 0 } { B 700 0 } 500 }
none
ROLE 4
defend_line { -1400 1150 } { -1400 -1150 } 1100 1400
none

Executing a Play

- Role assignment and role switching
- Sequencing tactics
- Termination

Playbooks

- What play is the right play?
  - The answer depends upon the opponent.
  - Use multiple alternative plans.

A Playbook

- What play do we select?
  - Use hand-chosen initial play weights.
  - Adapt play weights based on actual play outcomes.

Goals, penalties, free kicks, termination conditions, completed plays, timed out plays

Experts Advice

- Problem: Select which play to execute
- Similar to the "experts" on-line learning, k-armed bandits problems
- Select among a set of experts' advice
  - After selecting an expert, the payoffs of all the experts are revealed and the decision-maker appropriately rewarded.
  - The decision-maker performs selections repeatedly
  - Measure of performance: *regret*

Regret

- $r^t_{i=1\ldots n}$ be the reward at time t for expert i.
- x^t be the expert selected at time t.

Then the regret of the algorithm at time T is,

$$\textsc{Regret}^T = \max_{i=1\ldots n} \sum_{t=0}^{T} r_i^t - \sum_{t=0}^{T} r_{x^t}^t.$$

Regret is the amount of additional reward that could have been received if the algorithm had known which expert would be the best and chose that expert at each decision point.

Goal:

$$\lim_{T\to\infty} \frac{\textsc{Regret}^T}{T} = 0.$$

No-Regret Algorithm

Hedge, Auer et al. 95:

- At time T, choose expert i with the probability:

$$\Pr(x^T = i) = \frac{e^{R_i^{T-1}}}{\sum_{j=1}^{n} e^{R_j^{T-1}}},$$

where R_i^T is the cumulative reward received by expert i up to decision point T,

$$R_i^T = \sum_{t=0}^{T} r_i^t.$$

Plays as Experts' Advice

- Each play is an "expert" recommending a course of action.
- The team must select among its available plays, and after that play terminates, its outcome (e.g., goal, penalty kick, free kick, etc.) is the reward received for executing that play.
- A selection algorithm with zero-regret in the limit in the play context guarantees that the team over time is doing at least as well as its most effective play, (which is not known in advance.)

Play Selection Beyond Experts Task

Two complications preventing from the application of the standard no-regret experts Hedge algorithm:

- First, only the reward of the play selected is observed rather than the rewards of all of the plays that could have been selected.
- Second, not all plays are applicable at each decision point.

Addressing First Complication – Exp3

Exp3 – extension of Hedge, Hedge is run with:

$$\hat{r}_i^t = \begin{cases} r_i^t / Pr(x^t = i) & \text{if } x^t = i \\ 0 & \text{otherwise} \end{cases}.$$

Basic idea: observed rewards are scaled inversely with the probability of selecting that play. This allows an estimate of the total reward if the play had been chosen at each decision point.

Addressing Second Complication – Sleeping Experts

- Not all experts are available -- plays have applicability conditions.
- Sleeping experts - slight redefinition of regret, Freund et al. 97, to account for counting only the distribution of possible reward at each time step.

Modification to Hedge - if expert i is sleeping at time t,

$$Pr(x^{t+1} = i) = Pr(x^t = i),$$

that is, sleeping does not affect the expert's probability of being chosen in the future.

Sleeping Experts Version of Exp3

Let the weights be $w_i^t = e^{R_i^t}$.

Given that $R_i^T = \sum_{t=0}^{T} r_i^t$, note that adding r_i^t to R_i^t is the same as multiplying w_i^t by $e^{r_i^t}$. Let this factor be called the multiplier, or m_i^t.

Adapting Play Selection – The Details

- Play selection $$Pr(x^t = i) = \frac{a_i^t w_i^t}{\sum_j a_j^t w_j^t}.$$
- Weight adjustment... $$w_i^t = w_i^{t-1} (m_i^t)^{1/Pr(x^t = i)} \cdot N_i^t$$

where, $$N_i^t = \begin{cases} 1 & \text{if } a_i^t = 0 \\ \frac{\sum_j a_j^t w_j^{t-1}}{\sum_j a_j^t \hat{w}_j^t} & \text{otherwise} \end{cases}.$$

If the expert is sleeping, remains unchanged. If it is awake, then it is updated as with Exp3, but all the awake experts w_i^t's are normalized to have a constant sum.

Adapting Play Selection – An Example

- Play selection...

Play	w_i	Applicable	p_i
Offense 1	1	Yes	1/3
Offense 2	1	Yes	1/3
Corner Offense	1	Yes	1/3

Adapting Play Selection – An Example

- Play selection...

Play	w_i	Applicable	p_i
Offense 1	1	Yes	1/2
Offense 2	1	Yes	1/2
Corner Offense	1		0

- Offense 2 is selected and we score a goal – multiplier 1.5...

Play	w_i	Applicable	p_i
Offense 1	1	Yes	0.24
Offense 2	2.25	Yes	0.53
Corner Offense	1	Yes	0.24

Adapting Play Selection – An Example

- Play selection...

Play	w_i	Applicable	p_i
Offense 1	0.62	Yes	0.21
Offense 2	1.38	Yes	0.46
Corner Offense	1	Yes	0.33

Learning by Play Adaptation

- A playbook *(multiple experts)* is composed of multiple plays p_j with:
  - Applicability conditions, an associated weight w_j
  - Success, failure conditions, time limits
- Active play is selected as a function of the weights

$$P(p^* = p_j) = \frac{w_j}{\sum_{P_A} w_k}, \forall j \in P_A$$

Weights are adapted by feature observation.
Playbook adapts to the opponent
without explicit modeling.

Thinking… – Centralized "Brain"
(~every 17ms – 60Hz frame rate)

- ***Estimates state:***
  - ***processes current*** images, captured by two cameras, with 10 robots and ball, about 50 colored small blobs: pattern recognition, computation of position and orientation
  - ***predicts future*** positioning of 10 robots and ball for some future window
- ***Plans:***
  - ***for play:*** checks applicability, failure, timeout conditions of current play; if "play ended", evaluate state, prediction, goals, select new **play**
  - ***for path:*** uses models of robot dynamics (etc), positioning goals from play plus other path-specific goals – safety – and state, predictions, to select **tactic** parameters
- ***Acts:*** converts **skills** to commands to actuators

Multi-Robot Coordination, Planning

- Centralized perception and control
  - Excellent state estimation of the multi-robot system
  - Excellent play selection for teamwork
  - Learning through playbook adaptation
- Next: Distributed perception & control

Autonomous Robots

- Teams of 4 robots (initially 3 robots)
- Remarkable hardware - SONY AIBO robots
- Sensing, computing, and communication onboard
- Fully distributed - world modeling

Teamwork Without Communication

- Team is a set of *individual* robots
- View of the world solely from *own* sensors
- Teamwork achieved through *predefined roles*
  - Attacker*: "Can I see the ball? Go to the ball. Where am I? And where is the goal? Kick ball to goal."*
  - Goalie: *"Can I see the ball? Is the ball next to me? Clear the ball. Where am I? Go back to defend goal."*

Teamwork With Communication

- Team is still a set of *individual* robots
- Model of the world from *own* sensors **and** *communicated* information from team members.
  - Communication based on own sensing

State

- Own State – processed sensory data
  - "big" vector of task-relevant quantities:
  - ***Relative*** distance to task-relevant objects
    - Ball, goal, other robots, landmarks
- Sharing State – need localization
  - Position in *absolute* referential space for common "language"

Multi-Robot World Modeling

- Communication with latency
- Noise and confidence in shared information
- Multiple (variable) teammates

Challenge: Combine local and communicated information to form a coherent *world model*

Tracking Using Own Sensing

- Action models include probabilistic effects
- Effects are visited in order according to their probability

Distributed State Estimation *RMH: Ranked Multi-Hypothesis*

- Use *own* perception
- If object not in own view:
  - Generate a probabilistic set of hypotheses
    - Nondeterministic models of own actions
    - Teammate shared sensory data
    - …
  - Rank the hypotheses according to a confidence and utility function
  - Visit in order the ranked hypotheses

Summary - Distributed Planning

- Centralized control
  - Centralized evaluation and role assignment
  - Distributed bidding plus centralized role assignment
- Distributed control
  - Communication towards common world model
  - Distributed role assignment based on *common* world model