CS344M
Autonomous Multiagent Systems

Patrick MacAlpine

Department of Computer Science
The University of Texas at Austin

Good Afternoon, Colleagues

Are there any questions?

Logistics

- Project proposal questions?

Logistics

- Project proposal questions?
  - Hand in 2 hard copies, mark 2D/3D

Logistics

- Project proposal questions?
  - Hand in 2 hard copies, mark 2D/3D
  - Paper on pair programming

Logistics

- Project proposal questions?
  - Hand in 2 hard copies, mark 2D/3D
  - Paper on pair programming
- Next week's readings posted

Logistics

- Project proposal questions?
  - Hand in 2 hard copies, mark 2D/3D
  - Paper on pair programming
- Next week's readings posted
- Kim Houck RPE, Wednesday at 1, GDC 4.816
  - "Evolving Structure in Deep Neural Networks"

Motivation from real insects

- Ant colonies exhibit remarkably complex behaviors
  - Food gathering
  - Burial
  - Nest building
  - Reproduction

Motivation from real insects

- Ant colonies exhibit remarkably complex behaviors
  - Food gathering
  - Burial
  - Nest building
  - Reproduction
- Individual ants aren't smart
  - The complexity is in the environment (Simon)

Motivation from real insects

- Ant colonies exhibit remarkably complex behaviors
  - Food gathering
  - Burial
  - Nest building
  - Reproduction
- Individual ants aren't smart
  - The complexity is in the environment (Simon)
  - They're easily fooled out of their element (Feynman)

Motivation from real insects

- Ant colonies exhibit remarkably complex behaviors
  - Food gathering
  - Burial
  - Nest building
  - Reproduction
- Individual ants aren't smart
  - The complexity is in the environment (Simon)
  - They're easily fooled out of their element (Feynman)

Model the ant, not the colony

Go to the Ant

- Complex system behavior from many simple agents

Go to the Ant

- Complex system behavior from many simple agents
- Complexity comes from interactions, the environment

Agent Definition

Agents tied to environment

- *Agent = <State, Input, Output, Process>*

Agent Definition

$\boxed{\text{Agents tied to environment}}$

- *Agent = <State, Input, Output, Process>*
- *Environment = <State, Process>*

Agent Definition

$\boxed{\text{Agents tied to environment}}$

- *Agent = <State, Input, Output, Process>*
- *Environment = <State, Process>*

$\boxed{\text{Note: supports hierarchical agents}}$

Examples from Nature

- Ants: path planning

Examples from Nature

- Ants: path planning
- Ants: brood sorting

Examples from Nature

- Ants: path planning
- Ants: brood sorting
- Termites: nest building

Examples from Nature

- Ants: path planning
- Ants: brood sorting
- Termites: nest building
- Wasps: task differentiation

Examples from Nature

- Ants: path planning
- Ants: brood sorting
- Termites: nest building
- Wasps: task differentiation
- Birds and Fish: flocking

Examples from Nature

- Ants: path planning
- Ants: brood sorting
- Termites: nest building
- Wasps: task differentiation
- Birds and Fish: flocking
- Wolves: surrounding prey

Principles

- Try to avoid functional decomposition

Principles

- Try to avoid functional decomposition
- Simple agents (small, forgetful, local)

Principles

- Try to avoid functional decomposition
- Simple agents (small, forgetful, local)
- Decentralized control

Principles

- Try to avoid functional decomposition
- Simple agents (small, forgetful, local)
- Decentralized control
- System performance from interactions of many

Principles

- Try to avoid functional decomposition
- Simple agents (small, forgetful, local)
- Decentralized control
- System performance from interactions of many
- Diversity important: randomness, repulsion

Principles

- Try to avoid functional decomposition
- Simple agents (small, forgetful, local)
- Decentralized control
- System performance from interactions of many
- Diversity important: randomness, repulsion
- Embrace risk (expendability) and redundancy

Principles

- Try to avoid functional decomposition
- Simple agents (small, forgetful, local)
- Decentralized control
- System performance from interactions of many
- Diversity important: randomness, repulsion
- Embrace risk (expendability) and redundancy
- Agents should be able to share information

Principles

- Try to avoid functional decomposition
- Simple agents (small, forgetful, local)
- Decentralized control
- System performance from interactions of many
- Diversity important: randomness, repulsion
- Embrace risk (expendability) and redundancy
- Agents should be able to share information
- Mix planning with execution

Principles

- Try to avoid functional decomposition
- Simple agents (small, forgetful, local)
- Decentralized control
- System performance from interactions of many
- Diversity important: randomness, repulsion
- Embrace risk (expendability) and redundancy
- Agents should be able to share information
- Mix planning with execution
- Provide an "entropy leak"

Covering of Continuous Domains

- Simple, pheromone-based algorithm

Covering of Continuous Domains

- Simple, pheromone-based algorithm
- Provable properties

Covering of Continuous Domains

- Simple, pheromone-based algorithm
- Provable properties
  - Covers whole area in a finite time

Covering of Continuous Domains

- Simple, pheromone-based algorithm
- Provable properties
  - Covers whole area in a finite time
- Extensions
  - Repetitive coverage (continual area sweeping)

Covering of Continuous Domains

- Simple, pheromone-based algorithm
- Provable properties
  - Covers whole area in a finite time
- Extensions
  - Repetitive coverage (continual area sweeping)
  - Initial pheromone profile

Covering of Continuous Domains

- Simple, pheromone-based algorithm
- Provable properties
  - Covers whole area in a finite time
- Extensions
  - Repetitive coverage (continual area sweeping)
  - Initial pheromone profile
  - Multiple robots

Covering of Continuous Domains

- Simple, pheromone-based algorithm
- Provable properties
  - Covers whole area in a finite time
- Extensions
  - Repetitive coverage (continual area sweeping)
  - Initial pheromone profile
  - Multiple robots
  - Other metrics

Covering of Continuous Domains

- Simple, pheromone-based algorithm
- Provable properties
  - Covers whole area in a finite time
- Extensions
  - Repetitive coverage (continual area sweeping)
  - Initial pheromone profile
  - Multiple robots
  - Other metrics
- Experiments
  - Now multiple robots make a difference

Real Robot Applications

Trail-Laying Robots :

- An application to **real robots**
- Trails marked with a pen
- Also use simulations (video)

Real Robot Applications

Trail-Laying Robots :

- An application to **real robots**
- Trails marked with a pen
- Also use simulations (video)
  - Future options(?): odor, fluorescence

Real Robot Applications

Trail-Laying Robots :

- An application to **real robots**
- Trails marked with a pen
- Also use simulations (video)
  - Future options(?): odor, fluorescence

TERMES :

- Termite robots
- (video)