topics:

• today we will discuss multiagent-based simulation

software agents

an agent is...

• anything that can be viewed as perceiving its environment through sensors and acting upon that environment through effectors (e.g., motors)

(Esrau Russell & Peter Norvig, 2003)

• a system that is situated in an environment, and which is capable of perceiving its environment and acting in it to satisfy some objectives

(Michael Wooldridge, 2002)

different kinds of agents

• human “agent”:
  - environment: physical world
  - sensors: eyes, ears, . . .
  - effectors: hands, legs, . . .

• software agent:
  - environment: e.g., UNIX operating system
  - sensors: ls, ps, . . .
  - effectors: rm, chmod, . . .

• internet agent:
  - environment: the Internet
  - sensors: http requests
  - effectors: http commands

• embodied (robotic) agent:
  - environment: physical world
  - sensors: light meters, bumpers, thermometers, . . .
  - effectors: motors attached to wheels, legs, grippers, . . .

agent decision-making: what to do?

• need to know what to do in any given state
  - what = an action that the agent can take
  - state = a configuration of the agent and its environment
  - for example: the position of all the pieces on a chess board, or the robots and the ball on a robot soccer field, or the position of a robot’s gripper, or all the bids in an electronic market

• autonomy
  - a crucial concern for agents
  - run-time decisions are made by the agent alone—i.e., no human remote control
  - means behavior is based on own experience
  - implies learning, adaptation
multiagent system

- a multiagent system (MAS) is...

  an environment in which many (well, two or more) agents exist and interact

properties of multiagent systems

- individual agents are self-interested
  - i.e., they have their own goals, even though there may be team rewards for a group of agents achieving a goal together
- cooperation is not governed
  - it is emergent
  - (and is not necessarily a feature of every multiagent system)
- versus "distributed systems", where
  - goals are only group-based
  - cooperation is engineered to be inherent in the system

artificial life

- *Artificial Life as a Tool for Biological Inquiry*,
  by Charles Taylor and David Jefferson (1995)
- "ALife" consists of four levels:
  1. molecular level — "wetware"
  2. cellular level — "software"
  3. organism level — "hardware"
  4. population level — "multiagent systems"

ant systems

- *The Ant System: Optimization by a colony of cooperating agents*,
  by Marco Dorigo, Vittorio Maniezzo and Alberto Colorni (1996)
- the "ant system" as an approach to stochastic combinatorial optimization
- "ant" = simple agent which (sort of) mimics the behavior of real ants
  - real ants lay pheromone trails to mark their pathways
  - individual behavior is apparently random
  - collective behavior emerges as autocatalytic (⇒ positive feedback); i.e., the probability with which ants subsequently follow the same trail increases as more ants take that trail
- this is not about simulating ant colonies per se, but rather about taking advantage of abstract behavioral properties of ant colonies to address a class of combinatorial optimization problems
- major diversions from reality:
  - "ants" have memory
  - "ants" are not blind
“ants” live in a discretized environment (time and space)

- main experimental example: traveling salesman problem

ants move randomly at each point
- the distance between towns \(i\) and \(j\) is the Euclidean distance \(d_{i,j}\)
- ants are always in some town (apparently), so their “velocity” varies such that the distance between two towns is covered in one simulated time “step”...
- trails “evaporate” over time ⇒ change in intensity:
  \[ \Delta \tau_{i,j} \]
- transition probability is a trade-off between visibility and trail intensity
- tabu list keeps track of towns that have already been visited

three algorithmic variations were tested:
1. ant cycle:
   \[ \Delta \tau_{i,j}^{k} = \begin{cases} Q & \text{if ant } k \text{ goes from } i \text{ to } j \text{ between time } t \text{ and } t + 1 \\ 0 & \text{otherwise} \end{cases} \]
2. ant density:
   \[ \Delta \tau_{i,j}^{k} = \begin{cases} Q & \text{if ant } k \text{ goes from } i \text{ to } j \text{ between time } t \text{ and } t + 1 \\ 0 & \text{otherwise} \end{cases} \]
3. ant quantity:
   \[ \Delta \tau_{i,j}^{k} = \begin{cases} Q \cdot d_{i,j} & \text{if ant } k \text{ goes from } i \text{ to } j \text{ between time } t \text{ and } t + 1 \\ 0 & \text{otherwise} \end{cases} \]

experiments were conducted over many cycles and conditions
- a linear relationship was discovered between the number of towns and the number of ants
- strengths of approach:
  - good solutions found for all test problems within range of parameter optimality
  - algorithm converges quickly and doesn’t exhibit stagnation behavior
  - algorithm and parameter values appear relatively insensitive to increase in problem dimensions

multiagent-based simulation

- centralized versus decentralized models, ways of thinking
- the old way: centralized — “by lead or by seed”
- the new way: decentralized
- examples of decentralized computational models:
  - neural networks
  - subsumption architecture
  - cellular automata
- properties of decentralized models:
  - emergent behavior
  - evolutionary learning
multiagent simulation for learning

- *Modeling Nature’s Emergent Patterns with Multiagent Languages*,
  by Uri Wilensky (2002)
- decentralized tools for learning: *constructionism*
  - hands-on exploration
  - no recipe to follow
- **NetLogo**
  - “turtles (agents)"
  - “patches” (environment)
- lessons for understanding decentralized thinking:
  1. positive feedback isn’t always negative
  2. randomness can help create order
  3. a flock isn’t a big bird
  4. a traffic jam isn’t just a collection of cars
  5. the hills are alive