Lecture #7A
Distributed Problem Solving and Planning
- Distributed problem solving
- Overview of cooperation and coordination in agent systems
- Reference – Wooldridge Ch. 4, 8
- Reference – Weiss Ch. 3
- Reference – R & N Ch. 10,11
- Figures © MIT Press/Prentice Hall

Distributed Problem Solving
- Subfield of Artificial Intelligence
- Focus on solving problems that require collective effort (or may be more effectively solved with a distributed approach).

Example Problem
Towers of Hanoi

Working Together
- Why and how do agents work together?
- Important to make a distinction between:
  - benevolent agents and
  - self-interested agents.

Benevolent Agents
- If we “own” the whole system, we can design agents to help each other whenever asked.
- In this case, we can assume agents are benevolent: our best interest is their best interest.
- Problem-solving in benevolent systems is cooperative distributed problem solving (CDPS).
- Benevolence simplifies the system design task enormously!
Self-Interested Agents
- If agents represent individuals or organizations, (the more general case), then we cannot make the benevolence assumption.
- Agents will be assumed to act to further their own interests, possibly at expense of others.
- Potential for conflict.
- May complicate the design task enormously.

Cooperative Distributed Problem Solving
- Problem analysis and decomposition
- Sub problem solution
- Answer synthesis
- Factors
  - Sub problem independence?
  - Distributed?

Towers of Hanoi Layers

Result Sharing in Blackboard Systems
- The first scheme for cooperative problem solving: the blackboard system.
- Results shared via shared data structure (BB).
- Knowledge sources (KS).
- Multiple agents (KSS/KAs) can read and write to BB.
- Agents write partial solutions to BB.
- BB may be structured hierarchically.
- Mutual exclusion over BB required bottleneck.
- No concurrent activity.
Blackboard Architecture

Solution Design/Synthesis

- Decomposition
  - Hierarchical
  - Non-hierarchical

- Sub-problems
  - Independent – easier to combine to solve entire problem
  - Dependent/interactions – solution to parts must be combined or modified to solve entire problem

- Example: blocks world problems

Task Sharing and Result Sharing

- Two main modes of cooperative problem solving:
  - task sharing - components of a task are distributed to component agents
  - result sharing - information (partial results etc) is distributed and communicated between agents

Result Sharing Issues

- Confidence
  - Independently derived solutions can be cross-checked

- Completeness
  - Agents share local views to achieve better global view

- Precision
  - Agents share results to ensure that overall solution precision is increased

- Timeliness
  - Result Sharing produces more in faster problem-solving

The Contract Net

- Well known task-sharing protocol for task allocation is called contract net,

- 5 phases
  1. Recognition
  2. Announcement
  3. Bidding
  4. Awarding
  5. Expediting

Step 1 - Recognition

- In this stage, an agent recognizes it has a problem it wants help to achieve

- Agent has a goal, and either...
  - realizes it cannot achieve the goal in isolation — does not have capability;
  - realises it would prefer not to achieve the goal in isolation (typically because of solution quality, deadline, etc)
Step 2 - Announcement

- The agent sends out an announcement which includes a specification of the task to be achieved
- Specification must encode:
  - description of task itself (may be executable);
  - any constraints (e.g., deadlines, quality constraints)
  - meta-task information (e.g., “bids must be submitted by...”)
- The announcement is then broadcast

Step 3 - Bidding

- Agents that receive the announcement decide for themselves whether they wish to bid for the task
- Factors:
  - agent must decide whether it is capable of expediting task;
  - agent must determine quality constraints & price information (if relevant)
- If they do choose to bid, then they submit a tender

Awarding (4) and Expediting (5)

- Agent that sent task announcement must choose between bids & decide who to “award the contract” to
- The result of this process is communicated to agents that submitted a bid
- The successful contractor then expedites the task. May involve generating further manager-contractor relationships: sub-contracting

Implementing a Contract Net

- How to. . .
  - specify tasks?
  - specify quality of service?
  - select between competing offers?
  - differentiate between offers based on multiple criteria?

Result Sharing in Subscribe/Notify

- Common design pattern in OO systems: subscribe/notify
- An object subscribes to another object, saying “tell me when event e happens”
- When event e happens, original object is notified
- Information pro-actively shared between objects
- Objects required to know about the interests of other objects inform objects when relevant information arises
Summary

- Distributed Problem-Solving and Planning
- Overview of cooperation and coordination in agent systems