INF5390 – Kunstig intelligens
Planning and Acting

Roar Fjellheim
Outline

- Planning and scheduling
- Hierarchical task networks
- Conditional planning
- Monitoring and replanning
- Multi-agent planning
- Summary

AIMA Chapter 11: Planning and Acting in the Real World
Planning in ideal and real worlds

- Classical planners assume
  - Fully observable, static and deterministic domains
  - Correct and complete action descriptions
  - … allowing a “plan-first-then-act” planning approach

- … but in the real world
  - The world is dynamic, and time cannot be ignored
  - Information on the world is incomplete and incorrect
  - … the agent must be prepared for unexpected events

- Plus - scaling up to real-world problem size!
Time, schedules, and resources

- The PDDL language allows events (actions) and ordering of events, but not time duration.
- In real-life planning, we must take duration, delays, etc. into account (not just ordering).

Job shop scheduling:
- The problem is to complete a set of jobs.
- Each job consists of a set of actions, with given duration and resource requirements.
- Determine a schedule that minimizes total time (makespan) needed while respecting resource constraints.

- Must extend representation language to express duration and resource constraints.
Example - Assembling two cars

- Jobs({AddEngine1 < AddWheels1 < Inspect1},
  {AddEngine2 < AddWheels2 < Inspect2})
- Resources(EngineHoists(1), WheelStations(1),
  Inspectors(2), LugNuts(500))
- Action(AddEngine1, DURATION:30,
  USE: EngineHoists(1))
- Action(AddEngine2, DURATION:60,
  USE: EngineHoists(1))
- Action(AddWheels1, DURATION:30,
  CONSUME: LugNuts(20), USE: WheelStations(1))
- Action(AddWheels2, DURATION:15,
  CONSUME: LugNuts(20), USE: WheelStations(1))
- Action(InspectI, DURATION:10,
  USE: Inspectors(1))
Scheduling - No resource constraints

- Partial order plan produced by e.g. POP
- To create a schedule, we must place actions on a timeline
- Can use critical path method (CPM): the longest path, no slack – determines total duration
- Shortest duration schedule, given partial-order plan:

85 minutes
Scheduling with resource constraints

- Actions typically require resources
  - Consumable resources – e.g. LugNuts
  - Reusable resources – e.g. EngineHoists
- Resource constraints make scheduling more complex because of interaction between actions
- AI and OR (Operations Research) methods can be used to solve scheduling problems with resources
- Shortest duration gone up from 85 to 115 minutes
Planning and scheduling

The approach shown here is common in real-world AI applications for manufacturing scheduling, airline scheduling, etc.:

- First generate partial order plan without timing information (*planning*)
- Then use separate algorithm to find optimal (or satisfactory) time behavior (*scheduling*)

In some cases it may be better to *interleave* planning and scheduling, e.g. to consider temporal constraints already at the planning stage.
Reduce complexity by decomposition

- Often possible to reduce problem complexity by decompose to subproblems, solve independently, and assemble solution

- HTN – Hierarchical Task Networks
  - Planner keeps library of subplans
  - Extend planning algorithm to use subplans
  - Can reduce time & space requirements considerably

- Most real-world planners use HTN variants
Planning in nondeterministic domains

- **Nondeterministic worlds**
  - *Bounded* nondeterminism: Effects can be enumerated, but agent cannot know in advance which one will occur
  - *Unbounded* nondeterminism: The set of possible effects is unbounded or too large to enumerate

- **Planning for bounded nondeterminism**
  - Sensorless planning
  - Contingent planning

- **Planning for unbounded nondeterminism**
  - Online replanning
  - Continuous planning
Sensorless planning

- Agent has no sensors to tell which state it is in, therefore each action might lead to one of several possible outcomes
- Must reason about sets of states (belief states), and make sure it arrives in a goal state regardless of where it comes from and results of actions
- Nondeterminism of the environment does not matter – the agent cannot detect the difference anyway
- The required reasoning is often not feasible, and sensorless planning is therefore often not applicable
Contingent planning

- Constructs conditional plans with branches for each (enumerable) possible situation
- Decides which action to choose based on special sensing actions that become parts of the plan
- Can also tackle partially observable domains by including reasoning about belief states (as in sensorless planning)
- Planning algorithms have been extended to produce conditional branching plans
Online replanning

- Monitors situation as plan unfold, detects when things go wrong
- Performs replanning to find new ways to reach goals, if possible by repairing current plan

Agent proceeds from S, and next expects E following original whole-plan
- Detects that it’s actually in O
- Creates a repair plan that takes it from O to a state P in original plan
- New plan to reach G becomes repair + continuation
Contingent planning vs. replanning

- **Contingent planning**
  - ✓ All actions in the real world have additional outcomes
  - ✓ Number of possible outcomes grows exponentially with plan size, most of them are highly improbable
  - ✓ Only one outcome will actually occur

- **Replanning**
  - ✓ Basically assumes that no failure occurs
  - ✓ Tries to fix problems as they occur
  - ✓ May produce fragile plans, hard to fix if things go wrong
Continuous spectrum of planners

- Contingent planning and replanning are extremes of a spectrum, where intermediate solutions exist
  - Disjunctive outcomes for actions where more than one outcome is likely
  - Agent can insert sensing action to detect what happened and construct corresponding conditional plan
  - Other contingencies dealt with by replanning

- More generally, agents in complex domains and with incomplete/incorrect information should
  - Assess likelihood and costs of various outcomes
  - Construct plan that maximizes probability of success and minimizes cost
  - Ignore contingencies that are unlikely or easy to deal with
Continuous planning

- The planner persists over time – never stops, and interleaves planning, sensing and execution

- The continuously planning agent must
  - Execute steps of current plan (even if not complete)
  - Refine plan if not applicable or in conflict
  - Modify plan in light of new information
  - Formulate new goals when required

- Planners, e.g. partial-order planning (POP) can be extended to provide required functionality
Multi-agent planning

- Single-agent planning works against “nature”, but in many cases the environment includes other agents with their own goals
- Multi-agent environments can be
  - Cooperative: Agents work together to achieve some common goal
  - Competitive: Agents have conflicting goals
- Multi-agent architectures and applications, incl. planning, represent very active AI research area
Coordination of multi-agent planning

- Cooperative planning can produce *joint plans*
  - For each agent, the joint plan tells what to do
  - If each follows its plan, overall goal will be achieved

- Problems arise if several joint plans are possible
  - Each agent must know which plan to follow
  - Requires some form of *coordination*

- Coordination can be by
  - Convention or *social law*
  - Inter-agent *communication*
Summary

- Classical planning systems assume deterministic and static domains, and complete and correct information. Many domains violate this assumption.
- *Scheduling* is planning with time and resource constraints and are solved by special methods.
- Large planning problems can be made tractable by *hierarchic decomposition* (hierarchic task networks).
- Nondeterministic domains can be *bounded* (enumerable outcomes) or *unbounded* (any outcome is possible).
Summary (cont.)

- Planning in bounded determinism includes *sensorless planning* (make sure plan succeeds) or *contingent planning* (select one of multiple pre-made plans bases on sensing).
- Planning in unbounded nondeterminism includes *online replanning* (repair plan if failure) or *continuous planning* (ongoing adaptation of plan).
- *Multi-agent planning* applies to domains where there are other *cooperative* or *competitive* agents.