Artificial Intelligence Planning

Advanced Topics

Artificial Intelligence Planning

- **Advanced Topics**

Overview

- Before We Plan
- Plan Generation
- Scheduling (Resources)
- After We Plan

Overview

➢Before We Plan

•Plan Generation

•Scheduling (Resources)

•After We Plan

Knowledge Engineering

- problem formulation is vital for efficient problem-solving
- knowledge engineering phases (iterative process):
  - requirements specification
  - knowledge modelling
  - model analysis (verification and validation)
  - deploying the model (to the planner)
  - plan synthesis
  - plan-analysis and post-design
- learning domain models

Knowledge Engineering

- **problem formulation is vital for efficient problem-solving**
- **knowledge engineering phases (iterative process):**
  - **requirements specification**
    - elicitation, analysis and validation of requirements from domain experts
  - **knowledge modelling**
    - construction of a formal (human-understandable) model
  - **model analysis (verification and validation)**
    - using a domain expert
  - **deploying the model (to the planner)**
    - export model in formalism suitable for planner
  - **plan synthesis**
  - **plan-analysis and post-design**
    - analyse usability of plans (using metrics)
- **learning domain models**
  - automatically construct domain models

The Frame Problem

- problem: need to represent a long list of facts that are not changed by an action
- the <u>frame problem</u>: construct a formal framework for reasoning about actions and change in which the non-effects of actions do not have to be enumerated explicitly
- approaches:
  – use a different style of representation in first-order logic (same formalism)
  – use a different logical formalism, e.g. non-monotonic logic
  – write a procedure that generates the right conclusions and forget about the frame problem

The Frame Problem

•**problem: need to represent a long list of facts that are not changed by an action**

•example: extend domain with new relation; must examine all operators (elaboration tolerance)

•**the <u>frame problem</u>: construct a formal framework for reasoning about actions and change in which the non-effects of actions do not have to be enumerated explicitly**

•**approaches:**

•**use a different style of representation in first-order logic (same formalism)**

•different from original style, which was the situation calculus

•**use a different logical formalism, e.g. non-monotonic logic**

•**write a procedure that generates the right conclusions and forget about the frame problem**

•all the work described on this course falls under this

Overview

- Before We Plan
- Plan Generation
- Scheduling (Resources)
- After We Plan

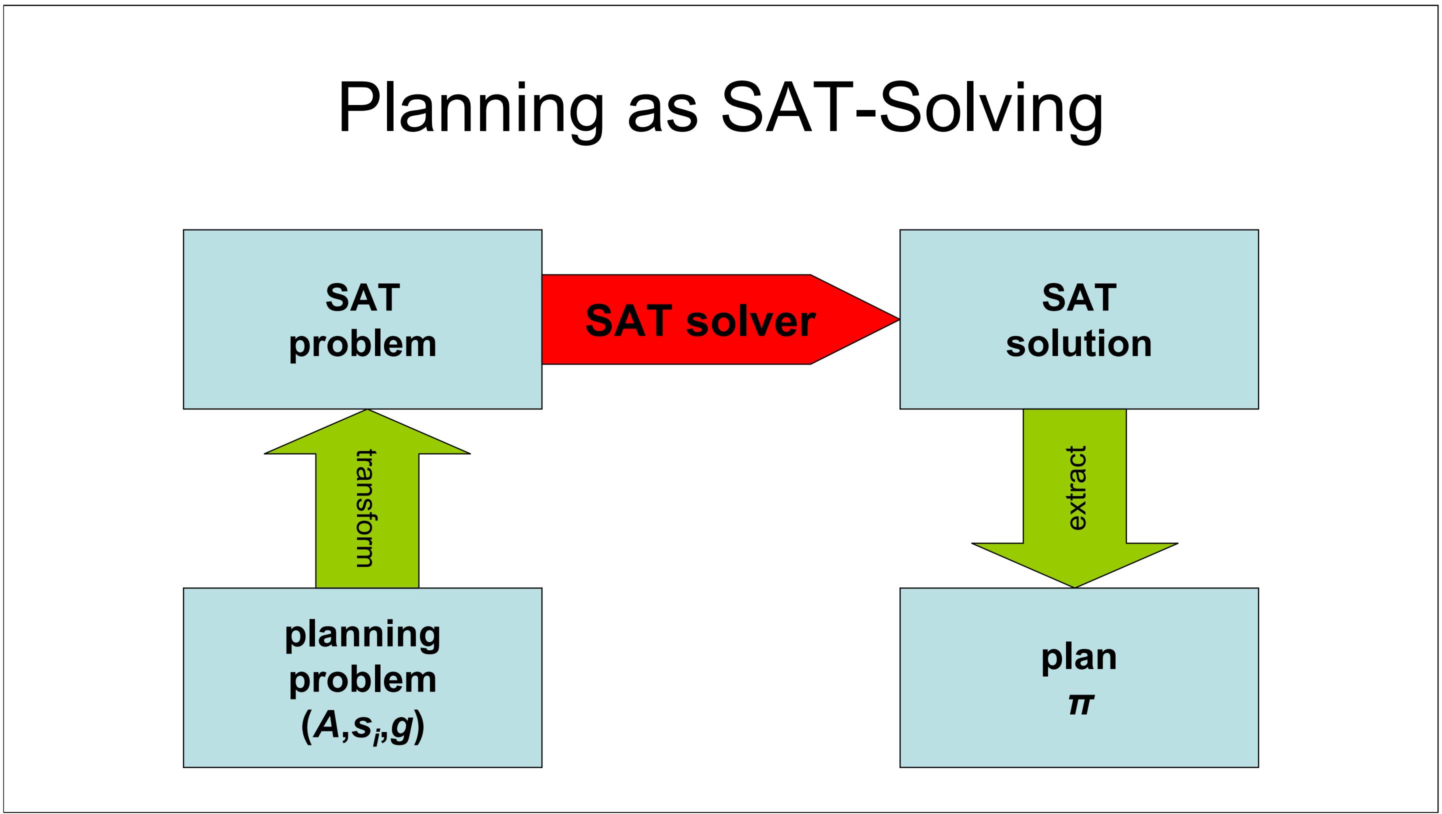
Overview

•Before We Plan

➢Plan Generation

•Scheduling (Resources)

•After We Plan



Planning as SAT-Solving

- **planning problem**
  - actually: bounded propositional planning problem
- **SAT problem**
  - set of propositional clauses
- **SAT solution**
  - assignment of truth values to propositions
- **plan**

Planning with Uncertainty

- problem: outcome of actions may be uncertain

- approach: belief state search
  - belief state: set of world states, one of which is actual state
  - solution plan is sequence of actions

- approach: contingency planning
  - represent possible outcomes of actions as contingencies
  - solution plan is a tree structure with observation actions

Planning with Uncertainty

- **problem: outcome of actions may be uncertain**
  - example: make a sandwich, but where is the knife?
- **approach: belief state search**
  - **belief state: set of world states, one of which is actual state**
    - initial state may also be belief state, e.g. knife in drawer or knife in sink
    - exponentially larger search space
  - **solution plan is sequence of actions**
- **approach: contingency planning**
  - **represent possible outcomes of actions as contingencies**
  - **solution plan is a tree structure with observation actions**

Probabilistic Planning

- Partially Observable Markov Decision Processes
  - set of world states S
  - set of actions A; applicable in $s \epsilon S$: $A(s) \subseteq A$
  - cost function: $c(a,s) > 0$ for $s \epsilon S$ and $a \epsilon A$
  - transition probabilities: $P_a(s'|s)$ for $s,s' \epsilon S$ and $a \epsilon A$

  - initial belief state (probability distribution over S)
  - final belief state

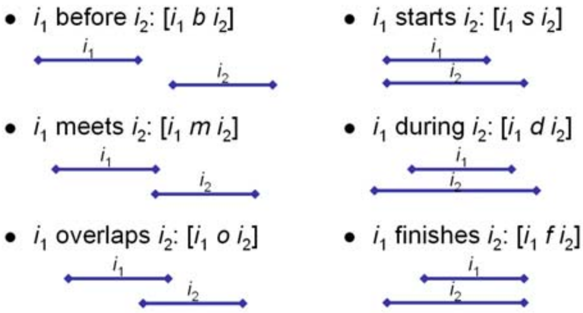
  - solution (policy): function from states to actions
  - optimal policy: minimal expected cost

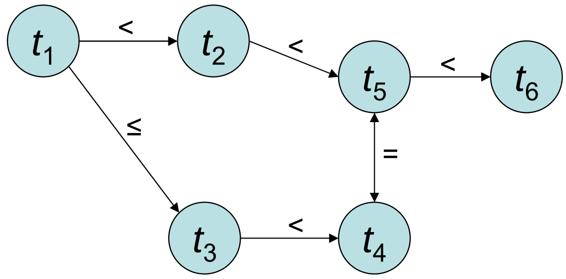
Probabilistic Planning

- **Partially Observable Markov Decision Processes**
  - **set of world states S**
  - **set of actions A; applicable in $s \epsilon S$: $A(s) \subseteq A$**
  - **cost function: $c(a,s) > 0$ for $s \epsilon S$ and $a \epsilon A$**
  - **transition probabilities: $P_a(s'|s)$ for $s,s' \epsilon S$ and $a \epsilon A$**
    - probability that we end up in s', given that we executed a in s
  - **initial belief state (probability distribution over S)**
    - may be 1 for single state if we know initial state
  - **final belief state**
  - **solution (policy): function from states to actions**
  - **optimal policy: minimal expected cost**

Planning with Time

- heuristic search: A* takes action cost into account
- time in partial plans (HTN with temporal constraints):
  – time point networks
  – interval algebra





Planning with Time

•durative actions: actions take (known amount of) time, have start and finish (time points)

•heuristic search: A* takes action cost into account

•MetricFF can plan with durative actions

•time in partial plans (HTN with temporal constraints):

•plan refinement asserts new constraints; network must remain consistent

•time point networks

•interval algebra

•more expressive than time point algebra

Learning to Plan (Better)

- general idea
  - let planner solve a series of (similar) planning problems
  - analyse problem-solving performed by planner
  - feed back analysis results into planning process
- learning macro-operations
- learning search control knowledge

Learning to Plan (Better)

•general idea

•let planner solve a series of (similar) planning problems

•analyse problem-solving performed by planner

•feed back analysis results into planning process

•learning macro-operations

•additional operators that combine existing operators

•idea similar to HTN planning

•learning search control knowledge

•learn rules that order nodes with the same f-value

Multi-Agent Planning

- problem: no single agent in control
  - agents with different beliefs
  - agents with different capabilities
  - agents with joint goal
  - agents with individual (conflicting) goals
- joint actions

Multi-Agent Planning

•assumption so far: single planner can use all available operators

•**problem: no single agent in control**

- •**agents with different beliefs**
  - •initial state not agreed; with or without contradictions; trust issues
- •**agents with different capabilities**
  - •DWR example: cranes and robots can do different things; cranes at different locations do different things
- •**agents with joint goal**
  - •single agreed goal for all agents; reward for agents?
- •**agents with individual (conflicting) goals**
  - •may not be solvable

•**joint actions**

- •example: lifting a table at both ends

Overview

- Before We Plan
- Plan Generation
- Scheduling (Resources)
- After We Plan

Overview

•Before We Plan

•Plan Generation

➢Scheduling (Resources)

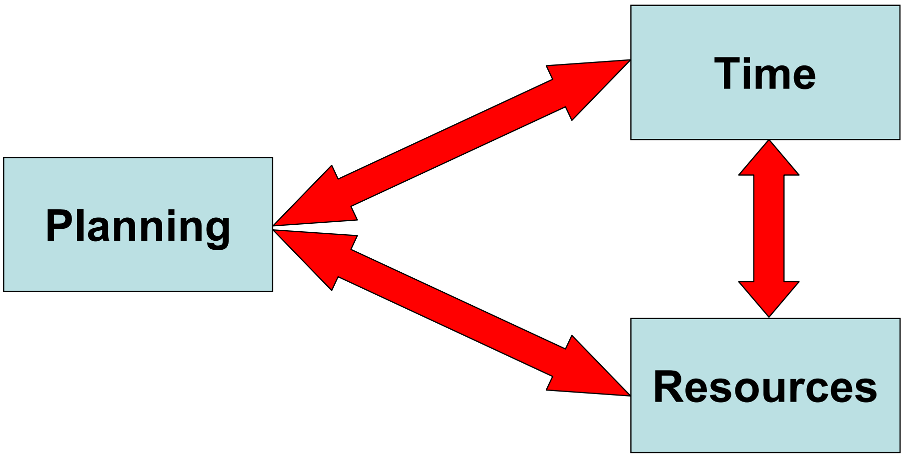
•After We Plan

Scheduling (Resources)

- resources: an entity needed to perform an action
  - state variables: modified by actions in absolute ways
  - resource variables: modified by actions in relative ways
- resource types:
  - reusable vs. consumable
  - discrete vs. continuous
  - unary
  - sharable
  - resources with states

- **Scheduling (Resources)**
- **resources: an entity needed to perform an action**
  - **state variables: modified by actions in absolute ways**
    - example: move(*r,l.l'*):
    - location changes from *l* to *l'*
  - **resource variables: modified by actions in relative ways**
    - example: move(*r,l.l'*):
    - fuel level changes from *f* to *f-f'*
- **reusable vs. comsumable**
- **discrete vs. continuous**
  - countable number of units: cranes, bolts
  - real-valued amount: bandwidth, electricity
- **unary**
  - Q_r=1; exactly one resource of this type available
- **sharable**
  - can be used by several actions at the same time
- **resources with states**
  - actions may require resources in specific state

Planning and Scheduling



Planning and Scheduling

- planning:
    - input: initial state, operators, goal
    - output: action sequence
- planning with time:
    - output: action sequence with start/finish times
    - time may invalidate out solution plans
- planning with resources:
    - output: action sequence with resources assigned to actions
    - resource availability may invalidate solution plans
- planning with time and resources:
    - not a sequential process!

Overview

- Before We Plan
- Plan Generation
- Scheduling (Resources)
- After We Plan

Overview

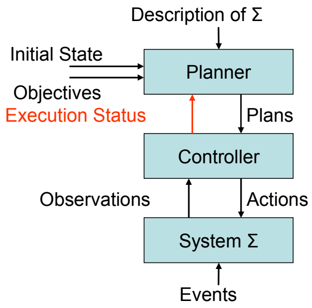
•Before We Plan

•Plan Generation

•Scheduling (Resources)

➢After We Plan

Plan Execution



- problem: real world differs from model described by Σ
- more realistic model: interleaved planning and execution
  – plan supervision
  – plan revision
  – re-planning
- dynamic planning: closed loop between planner and controller
  – execution status

Plan Execution

•problem: physical system differs from model described by Σ

•planner only has access to model (description of Σ)

•controller must cope with differences between Σ and real world

•more realistic model: interleaved planning and execution

•**plan supervision**: detect when observations differ from expected results

•**plan revision**: adapt existing plan to new circumstances

•**re-planning**: generate a new plan from current (initial) state

•dynamic planning: closed loop between planner and controller

•**execution status**

Multiple Agents

- coordination
  - ordering constraints between actions assigned to different agents
  - actions with shared (limited) resources
  - joint actions
- communication
  - plans may involve communication actions (for coordination)
  - observation actions: communicate results
- execution failure recovery
  - local plan repair
  - propagation of changes to the plan (de-commitment)

Multiple Agents

•coordination

•ordering constraints between actions assigned to different agents

•actions with shared (limited) resources

•joint actions

•communication

•plans may involve communication actions (for coordination)

•observation actions: communicate results

•execution failure recovery

•local plan repair

•propagation of changes to the plan (de-commitment)

Overview

- Before We Plan
- Plan Generation
- Scheduling (Resources)
- After We Plan

Overview
•Before We Plan
•Plan Generation
•Scheduling (Resources)
•After We Plan