Automated Planning

Introduction and Overview

Literature


Overview

- What is AI Planning?
  - A Conceptual Model for Planning
  - Restricting Assumptions
  - A Running Example: Dock-Worker Robots

Human Planning and Acting

- acting without (explicit) planning:
  - when purpose is immediate
  - when performing well-trained behaviours
  - when course of action can be freely adapted
- acting after planning:
  - when addressing a new situation
  - when tasks are complex
  - when the environment imposes high risk/cost
  - when collaborating with others

- people plan only when strictly necessary
Defining AI Planning

- **planning**:
  - explicit deliberation process that chooses and organizes actions by anticipating their outcomes
  - aims at achieving some pre-stated objectives

- **AI planning**:
  - computational study of this deliberation process

Why Study Planning in AI?

- **scientific goal of AI**: understand intelligence
  - planning is an important component of rational (intelligent) behaviour

- **engineering goal of AI**: build intelligent entities
  - build planning software for choosing and organizing actions for autonomous intelligent machines
Domain-Specific vs. Domain-Independent Planning

- **domain-specific planning**: use specific representations and techniques adapted to each problem
  - important domains: path and motion planning, perception planning, manipulation planning, communication planning
- **domain-independent planning**: use generic representations and techniques
  - exploit commonalities to all forms of planning
  - leads to general understanding of planning

- domain-independent planning complements domain-specific planning

Overview

- What is AI Planning?
- A Conceptual Model for Planning
- Restricting Assumptions
- A Running Example: Dock-Worker Robots
Why a Conceptual Model?

- conceptual model: theoretical device for describing the elements of a problem
- good for:
  - explaining basic concepts
  - clarifying assumptions
  - analyzing requirements
  - proving semantic properties
- not good for:
  - efficient algorithms and computational concerns

Conceptual Model for Planning: State-Transition Systems

- A state-transition system is a 4-tuple \( \Sigma = (S,A,E,\gamma) \), where:
  - \( S = \{s_1, s_2, \ldots\} \) is a finite or recursively enumerable set of states;
  - \( A = \{a_1, a_2, \ldots\} \) is a finite or recursively enumerable set of actions;
  - \( E = \{e_1, e_2, \ldots\} \) is a finite or recursively enumerable set of events; and
  - \( \gamma: S \times (A \cup E) \rightarrow 2^S \) is a state transition function.
- if \( a \in A \) and \( \gamma(s,a) \neq \emptyset \) then \( a \) is applicable in \( s \)
- applying \( a \) in \( s \) will take the system to \( s' \in \gamma(s,a) \)
State-Transition Systems as Graphs

- A state-transition system $\Sigma = (S,A,E,\gamma)$ can be represented by a directed labelled graph $G = (N_G,E_G)$ where:
  - the nodes correspond to the states in $S$, i.e. $N_G = S$, and
  - there is an arc from $s \in N_G$ to $s' \in N_G$, i.e. $s \rightarrow s' \in E_G$, with label $u \in (A \cup E)$ if and only if $s' \in \gamma(s,a)$.

State-Transition Graph Example: Missionaries and Cannibals
Objectives and Plans

- state-transition system:
  - describes all ways in which a system may evolve
- plan:
  - a structure that gives appropriate actions to apply in order to achieve some objective when starting from a given state
- types of objective:
  - goal state $s_g$ or set of goal states $S_g$
  - satisfy some conditions over the sequence of states
  - optimize utility function attached to states
  - task to be performed

Planning and Plan Execution

- planner:
  - given: description of $\Sigma$, initial state, objective
  - generate: plan that achieves objective
- controller:
  - given: plan, current state (observation function: $\eta: S \rightarrow O$)
  - generate: action
- state-transition system:
  - evolves as actions are executed and events occur
Dynamic Planning

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

Overview

- What is AI Planning?
- A Conceptual Model for Planning
  - Restricting Assumptions
- A Running Example: Dock-Worker Robots
**A0: Finite \( \Sigma \)**

- **Assumption A0**
  - system \( \Sigma \) has a finite set of states
- **Relaxing A0**
  - **why?**
    - to describe actions that construct or bring new objects into the world
    - to handle numerical state variables
  - **issues:**
    - decidability and termination of planners

**A1: Fully Observable \( \Sigma \)**

- **Assumption A1**
  - system \( \Sigma \) is fully observable, i.e. \( \eta \) is the identity function
- **Relaxing A1**
  - **why?**
    - to handle states in which not every aspect is or can be known
  - **issues:**
    - if \( \eta(s)=o \), \( \eta^{-1}(o) \) usually more than one state (ambiguity)
    - determining the successor state
A2: Deterministic $\Sigma$

- Assumption A2
  - system $\Sigma$ is deterministic, i.e. for all $s \in S$, $u \in A \cup E$:
    \[ |\gamma(s,u)| \leq 1 \]
  - short form: $\gamma(s,u) = s'$ for $\gamma(s,u) = \{s'\}$
- Relaxing A2
  - why?
    - to plan with actions that may have multiple alternative outcomes
  - issues:
    - controller has to observe actual outcomes of actions
    - solution plan may include conditional and iterative constructs

A3: Static $\Sigma$

- Assumption A3
  - system $\Sigma$ is static, i.e. $E = \emptyset$
  - short form: $\Sigma = (S,A,\gamma)$ for $\Sigma = (S,A,\emptyset,\gamma)$
- Relaxing A3
  - why?
    - to model a world in which events can occur
  - issues:
    - world becomes nondeterministic from the point of view of the planner (same issues)
A4: Restricted Goals

- Assumption A4
  - the planner handles only restricted goals that are given as an explicit goal state $s_g$ or set of goal states $S_g$

- Relaxing A4
  - why?
    - to handle constraints on states and plans, utility functions, or tasks
  - issues:
    - representation and reasoning over constraints, utility, and tasks

A5: Sequential Plans

- Assumption A5
  - a solution plan is a linearly ordered finite sequence of actions

- Relaxing A5
  - why?
    - to handle dynamic systems (see A3: static $\Sigma$)
    - to create different types of plans
  - issues:
    - must not shift problem to the controller
    - reasoning about (more complex) data structures
A6: Implicit Time

- **Assumption A6**
  - actions and events have no duration in state transition systems

- **Relaxing A6**
  - why?
    - to handle action duration, concurrency, and deadlines
  - issues:
    - representation of and reasoning about time
    - controller must wait for effects of actions to occur

A7: Offline Planning

- **Assumption A7**
  - planner is not concerned with changes of $\Sigma$ while it is planning

- **Relaxing A7**
  - why?
    - to drive a system towards some objectives
  - issues:
    - check whether the current plan remains valid
    - if needed, revise current plan or re-plan
The Restricted Model

- restricted model: make assumptions A0-A7

- Given a planning problem $\mathcal{P} = (\Sigma, s_i, S_g)$ where
  - $\Sigma = (S, A, \gamma)$ is a state transition system,
  - $s_i \in S$ is the initial state, and
  - $S_g \subset S$ is a set of goal states,
- find a sequence of actions $\langle a_1, a_2, \ldots, a_k \rangle$
  - corresponding to a sequence of state transitions $\langle s_i, s_1, \ldots, s_k \rangle$ such that
    - $s_1 = \gamma(s_i, a_1)$, $s_2 = \gamma(s_1, a_2)$, ..., $s_k = \gamma(s_{k-1}, a_k)$, and $s_k \in S_g$.

Restrictedness?

- non-deterministic state-transition system:
- equivalent deterministic state-transition system:
Overview

- What is AI Planning?
- A Conceptual Model for Planning
- Restricting Assumptions

A Running Example: Dock-Worker Robots

The Dock-Worker Robots (DWR) Domain

- aim: have one example to illustrate planning procedures and techniques
- informal description:
  - harbour with several locations (docks), docked ships, storage areas for containers, and parking areas for trucks and trains
  - cranes to load and unload ships etc., and robot carts to move containers around
**Automated Planning: Introduction and Overview**

---

**DWR Example State**

![Diagram of DWR Example State]

---

**Actions in the DWR Domain**

- **move** robot \( r \) from location \( l \) to some adjacent and unoccupied location \( l' \)
- **take** container \( c \) with empty crane \( k \) from the top of pile \( p \), all located at the same location \( l \)
- **put** down container \( c \) held by crane \( k \) on top of pile \( p \), all located at location \( l \)
- **load** container \( c \) held by crane \( k \) onto unloaded robot \( r \), all located at location \( l \)
- **unload** container \( c \) with empty crane \( k \) from loaded robot \( r \), all located at location \( l \)
State-Transition Systems: Graph Example

Overview

- What is AI Planning?
- A Conceptual Model for Planning
- Restricting Assumptions
- A Running Example: Dock-Worker Robots