Application of Artificial Intelligent Planning

by

Komminist (04305201)
Somasekara Rao (05407008)

November 13, 2005
Motivation

AI Planning is ubiquitous in on a daily basis life, for example, from planning how to make dinner to planning how to graduate from University with the least amount of work. Researchers in AI community have studied AI planning problems for many decades, and many techniques exist for automating planning processes. This seminar explore both conventional and modern approaches to AI planning. Issues to be discussed include: how to represent actions and world state, how to search for plans efficiently, and more .. by the help of example(s)
Outline

- Introduction
- Need of AI Planning
- Properties of AI Planning
- Types of Planning Problem and Algorithms
- Planning in Application
- Summary and Conclusions
Definition Of AI planning

Planning everywhere ...

- Planning is a problem solving technique.
- Planning is reasoning about future events in order to verify the existence of a reasonable series of actions to take in order to accomplish a goal.

...we do planning ...

- to reduce searching,
- to resolve conflicts, and
- to provide basis for error recovery.
AI Planning ...

- In more general sense: Given
  - a way to describe the world
  - an initial state of the world
  - a set of goals to achieve
  - a set of possible action to change the world

- Find
  - A prescription for actions to change the initial state into one that fulfill the goal
As actions change the world or we consider possible actions, we want to:

- Know how an action change the world,
- Keep track of the history of the world states (have we been here before?)
- Answer questions about potential world states (what would happened if...?)
Web management agents:
- *An Efficient Plan Execution System for Information Management Agents for gathering information from the World Wide Web...*

Robot planning:
- *For controlling the legs, arms and action of the robot (perhaps most of planning research relay here!)*

Manufacturing planning, Image processing management, Linguistic transportation

by Komminist (04305201) Somasekara Rao (05407008)
Crisis management, Bank risk management, Blocks-world, Puzzles, Artificial Domain, Software planning (postponed example later) and more...
Properties of Planning Algorithms

- **Soundness**
  - A planning algorithm is **sound** if all solutions found are legal plans
    - All preconditions and goals are satisfied
    - No constraints are violated (temporal, variable binding)

- **Completeness**
  - A planning algorithm is **complete** if a solution can be found whenever one actually exists
  - A planning algorithm is **strictly complete** if all solutions are included in the search space

- **Optimality**
  - A planning algorithm is **optimal** if the order in which solutions are found is consistent with some measure of plan quality
Types of Planning

- **Linear planning** - Planning with a goal stack.
  - Work on one goal until completely solved before moving on to the next goal.
  - More feasible if zero goal interaction!

- **Nonlinear planning** - Handles goal interactions by Interleaving.
  - Use goal set instead of goal stack
  - Include in the search space all possible subgoal orderings
**Linear planning: discussion ...**

- **Basic Idea**
  - Work on one goal until *completely* solved before moving on to the next goal

- **Planning algorithm maintains** goal Stack

- **Implications**
  - No *interleaving* of goal achievement
  - Efficient search if goals *do not* interact much !!!
Means-Ends Analysis

- **Basic Idea**
  - Search only relevant aspects of problem
  - What *means (operators)* are available to achieve the desired *ends* goal

- Find **difference** between goal and current state
- Find **operator** to reduce difference
- Perform **means-ends** analysis on new subgoals ...
- Used often in **Agent-Oriented Software Modelling language**, such as *i**, Tropos,*...
GPS (Newell, Simon, Ernst, 1960s)

- Introduced concept of means-ends analysis
- Essentially linear planning using recursive procedure calls as the goal-stack mechanism
- Simple and easy as well as the first planning algorithm.
STRIPS (Fikes, Nilsson, 1971)

- Same basic idea as GPS, but
- Solved the frame problem (STRIPS assumptions)
  - Actions must specify all the state variables whose values they change...
  - No disjunction allowed in effects and like that ...
- Introduced operator representation
- Operationalized notion of difference, subgoals, and operator application
- Dealt (somewhat) with plan execution and learning.
Pros and Cons on Linearity

**Advantages**
- Reduced search space, since goals are solved one at a time.
- Advantageous if goals are (mainly) independent
- Linear planning is **sound**

**Disadvantages**
- Linear planning may produce **suboptimal solutions** (based on the number of operators in the plan)
- Linear planning is **incomplete**.
Example: One-way Rocket

(Op : LOAD
 prec
 At(obj,loc)
 At(roc,loc)
 effects
 add : inside(obj,roc)
 del : At(obj,loc)

Op : UNLOAD
 prec
 inside(obj,roc)
 At(roc,loc)
 effects
 add : At(obj,loc)
 del : inside(obj,roc)

Op : MOVE
 prec
 At(roc,from)
 has-fuel(roc)
 effects
 add : At(roc,to)
 del : At(roc,from),
 del : has-fuel(roc)
Incompleteness of Linear Planning

Initial state:
At(obj1,locA)
At(obj2, locA)
At(roc, locA)
has-fuel(roc)

Goal Statement:
( and At(obj1,locB)
At(obj2, locB )

<table>
<thead>
<tr>
<th>Goal</th>
<th>Plan</th>
</tr>
</thead>
<tbody>
<tr>
<td>At(obj1,locB)</td>
<td>LOAD-ROCKET obj1 locA)</td>
</tr>
<tr>
<td></td>
<td>(MOVE-ROCKET)</td>
</tr>
<tr>
<td></td>
<td>(UNLOAD-ROCKET obj1 locb)</td>
</tr>
<tr>
<td>At(obj2,locB)</td>
<td>failure</td>
</tr>
</tbody>
</table>
**State-Space Search**

- **Node** denoted a state of the world.
- **Arcs** corresponds to the execution of a specific action.
- Planning problem: finding a path.
- **Difficulty**: requires Complete Description of the states searched, and requires the search to be carried out locally.
- Can be progression or regression ...
Plan-space search

- **Each node** in the graph represents partial plans.
- **Arcs** denote plan refinement operations.
- One can search for a plan: total order planning, or partial order planning.
Partial Order Planning

**Basic Idea:**
- Search in plan space and use least commitment, when possible

**Initial Plan:**
- \(< \{start, finish\}, \{start < finish, \{\}\}\)\>
- *start* has **preconditions**; Its effects are the initial state
- *finish* has **no effects**; Its preconditions are the goals
A set of actions: For example,

\{eat-breakfast \ (go to mess), take-shower, wake-up, go-to-AI Seminar\}

A set of ordering constraints: For example,

\{wake-up before eat-breakfast, wake-up before take-shower, wake-up before go-to-AI Seminar, take-shower before go-to-work\}

A set of causal links (set of bindings): For example,

wake-up $\rightarrow$ awake $\rightarrow$ eat-breakfast

is a link from the action, wake-up to the action eat-breakfast.

Note: When the action wake-up is added to the plan, the above causal link is recorded, along with the ordering constraint \{wake-up before eat-breakfast\}
Least Commitment

- **Basic Idea**
  - Make choices only that are relevant to solving the current part of the problem

- **Least commitment choices**
  - Orderings: Leave actions unordered, unless they must be sequential
  - Bindings: Leave variables unbounded, unless needed to unify with conditions being achieved.
  - Actions: Usually not subjected to "least commitment"

- **Refinement**
  - Only **add** information to the current plan
  - Transformational planning can remove choices
Many more non-linear planning

- **Graph-based search**
  - Sat-based search
  - OBDD-based search

- **Hierarchical planning**
  - Emphasis on action decomposition/refinement
  - Knowledge engineering/acquisition
  - Very little search

- many more ...
Classical Deterministic planning

- State of the world at any moment is **unambiguously** determined by the initial state of the world and **sequence of actions** that have been taken.

- **Action Model**: Complete, deterministic, correct, rich representation

- **State**: single initial state, fully known

- **Goals**: complete satisfaction
Deterministic planning contd ...

- Given set of states = **black dots**, actions = (blue, red) and initial state = I and a set of goal states = G
- The task is to find the a **path** from initial state to one of the goal states........
Non-deterministic Planning

- The world is **not always deterministic in nature***!!! Why ...?
- At least for two reasons:
  - The model of the World is usually very incomplete!
    - *We do not know whether somebody is going to phone or visit us, and then the visit or phone call can be modeled as a nondeterministic event that may or may not take place.*
  - Second, many actions themselves are by their nature nondeterministic, either intentionally or unintentionally.
    - *Throwing two dice has 12 possible outcomes that usually cannot be predicted (which is why throwing dice is interesting!)*
    - Throwing some object to a garbage bin from a distance may or may not succeed. Try it after this !!!

by Komminist (04305201) Somasekara Rao (05407008)

Application of Artificial Intelligent Planning
we added a red arrow to the second state above the initial state,

as well as a new state to the bottom of right corner with a nondeterministic transition leading to it.

the new arrows make the red action nondeterministic, and an impact on which action sequences are plans.
State - Space Nonlinear Planning

- **Extend linear planning**
  - Use goal set instead of goal stack
  - Include in the search space all possible subgoal orderings
  - Handles goal interactions by interleaving

- **Advantages**
  - Non-linear planning is sound
  - Non-linear planning is complete
  - Non-linear planning may be optimal with respect to plan length (depending on search strategy employed)

- **Disadvantages**
  - Larger search space, since all possible goal orderings may have to be considered
  - Somewhat more complex algorithm; More bookkeeping
Completeness of State-Space Nonlinear Planning

State-space nonlinear planning is **Complete**
(Continued from the incompleteness of the linear planning ...)

<table>
<thead>
<tr>
<th>Goal</th>
<th>Plan</th>
</tr>
</thead>
<tbody>
<tr>
<td>At(obj1,locB)</td>
<td>(LOAD obj1 locA)</td>
</tr>
<tr>
<td>At(obj2,locB)</td>
<td>(LOAD obj1 locA)</td>
</tr>
<tr>
<td>At(obj1,locB)</td>
<td>(MOVE)</td>
</tr>
<tr>
<td></td>
<td>(ULOAD obj1 locB)</td>
</tr>
<tr>
<td>At(obj2,locB)</td>
<td>(ULOAD obj2 locB)</td>
</tr>
</tbody>
</table>
**Generation a Solution plan**

- **A complex process:**
  - Alternative operators to achieve a goal.
  - Multiple goals that interacts
  - Efficiency and plan quality.
More Planning under nonlinear

- **NOAM** [Sacerdoti 1975]
  - First nonlinear, partial-order planner
  - Introduce the notion of plan-space search
  - Used TOME (Table Of Multiple Effects) to detect goal interactions

- **SNLP** (Systematic Non Linear Planner), **UCPOP** (Universal Conditional POP)

- **SHOP2** (Simple Hierarchical Order Planner)

- **SAVTA** (Subgoal After Every Try to Apply), **SABA** (Subgoal Always Before Applying)
Planning for AOSE: Motivation to i*

- i*: graphical modeling language used in AOSE (tropos)
- \( i* \rightarrow \) a framework for modeling social settings, based on the notions of actors, goals, dependencies (social interactions)...
- i* framework consists of two graphical models: SD and SR
- The SD diagram is used to represent the strategic dependencies.
  - Which express intentional relationships that exist among actors in order to fulfill some strategic objectives.
- Dependency type: goal Dependency, Softgoal Dependency, Task Dependency, Resource Dependency
Applying AI Planning for SE
AI Planning for Tool construction

SD: for the CAS

Dependency type

Internal goal

Actor

by Komminist (04305201) Somasekara Rao (05407008)
SR provides a more detailed level of modeling than SD by looking *in-side* actor.
So where is AI planning used?

- Goal decomposition/refinement (Conjunction and/or disjunction) ...
  - GetAdmission: (Apply, PassExam, GetAdmResult)

- Ordering (partial or total) should has to consider (prior-to):
  - For example
    - Apply is possible iff you have pass exam!!

- Imagin that there is high goal interaction among actors via dependencies!!
  - To get admission a student needs admission co-ordinator which is the RAO

- So here is the challenge!! But carefully choose your planner!
Example-1

Initial State

Set Of Objects

- part1, type: PART
- drill-1, type: SPOT-DRILL
- drill-2, type: TWIST-DRILL

Goal Statement

has-hole(part)
**Example 1 Contd..**

- **drill-spot**(part, drill-bit)
  - Pre: (holding-tool drill-bit) (holding-part part)
  - Add: (has-spot part)

- **drill-hole**(part, drill-bit)
  - Pre: (has-spot part) (holding-tool drill-bit) (holding-part part)
  - Add: (has-hole part)

- **put-drill-bit**(drill-bit)
  - Pre: (tool-holder-empty)
  - Add: (holding-tool drill-bit)
  - Del: tool-holder-empty

- **put-part**(part)
  - Pre: part-holder-empty
  - Add: (holding-part part)

- **remove-drill-bit**(drill-bit)
  - Pre: (holding-tool drill-bit)
  - Add: tool-holder-empty
  - Del: (holding-tool drill-bit)

- **remove-part**(part)
  - Pre: (holding-part part)
  - Add: part-holder-empty
Example-1 contd… sequence of actions

- put-part(part)
- put-drill-bit(drill-1)
- drill-spot(part drill-1)
- remove-drill-bit(drill-1)
- put-drill-bit(drill-2)
- drill-hole(part drill2)
Applications: Current and Potential

- Scheduling problems with action choices as well as resource handling requirements
  - Problems in supply chain management
  - HSTSS (Hubble Space Telescope scheduler)
  - Workflow management
- Autonomous agents
  - RAX/PS (The NASA Deep Space planning agent)
- Software module integrators
  - Test case generation (Pittsburgh)
- Interactive decision support
  - Monitoring subgoal interactions
- Plan-based interfaces
Current research direction

- Planning application in **Robotics**
- Planning for **web-service composition**
- Planning **databases**
- **Distributed Intelligent Agents Group**: For example
  - Multiagent Planning and Coordination as Distributed Search
  - Multiagent Planning and Coordination for Unmanned Ground Vehicles
  - Intelligent Agent Infrastructures for Supporting Collaboration
- more description =>
  [http://ai.eecs.umich.edu/diag/homepage.html](http://ai.eecs.umich.edu/diag/homepage.html)
- **AI Planning in Grid Computing**
- **AI planning in Tropos Methodology** and more .....
No general planning algorithm

Tackling real-world planning problems often requires considering various types of constraints, ranging from simple numerical comparators to complex resources.

There is a method how to solve planning tasks within general constraint-solving frameworks, such as propositional satisfiability, integer programming, and constraint programming. "paper: Constraints and AI Planning"

Planning as search: control rules to capture heuristics for efficient search; learning opportunities!
Conclusion contd ...

- Applying AI planning to **Early-phase of SE** ensures **quality** of Software Development in both functional and non-functional aspect of the system.

- For example, Tropos Software engineering use planning for searching of interactive goal in the specification to **handle consistencies** among different actors hence which is more amendable for formal verification and specification of softwares.

- **Every body has to practice planning to become rich ! (Please do planning now onwards!!)**
References and Links

- AIPS 2002 Sixth International conference on Artificial Intelligence Planning and Scheduling (AIPS2002) Workshop on Knowledge Engineering Tools and Techniques for AI Planning Toulouse, France 24 April 2002


Finding plans and defining plan constraints simultaneously”
Jim Blythe KCAP 01 Workshop on Interactive Tools for 
Knowledge Capture, 2001

Thomas A. Montgomery and Edmund H. Durfee, “Search 
Reduction in Hierarchical Distributed Problem Solving”. 
Group Decision and Negotiation 2:301-317 (Special issue on 
Distributed Artificial Intelligence), 1993.

http://www.cs.cmu.edu/ reids/planning/notes.html ( Date : 
10 / 10 / 2005 )
References and Links Contd ...

- http://www.planet-noe.org (Date: 23/10/2005)
- http://www.informatik.uni-freiburg.de/ki/lehre/ss04/aip/ (Date 10/10/2005)
Thank You