Outline

- Kitchen domain
- Planning bias and strategy
- Multi-strategy planning example (Case-, macro, and abstraction-based planning)
- DoLittle design
- Results
- Conclusion

Strategical Planning

- Early, popular research area in AI
- Problem
  - Input: initial state, set of actions, goal
  - Output: sequence of actions
- Intractable even for simple domains (blocksworld)
- No general problem solver (GPS)
Kitchen Domain

- One-armed mobile robot
- Simulation
- Low-level behaviors can be implemented
- Tasks: prepare hot/cold beverages:
  - Tea with milk
  - (Instant) coffee with sugar
  - Milk with honey

---

Make Tea (Part 1)

1. Open-door(cupboard)
2. Pick-up-from-cupboard(cup1)
3. Move-robot(at-table-at-sink)
4. Put-in-sink(cup1)
5. Fill-with-water(cup1)
6. Turn-water-off
7. Pick-up-from-sink(cup1)
8. Move-robot(at-sink-at-table)
9. Put-on-table(cup1)
### Make Tea (Part 2)

10. Move-robot(above-table at-stove) ; open microwave door
11. Open-door(microwave)
12. Move-robot(above-table at-stove)
13. Pick-up-from-table(cup1) ; put cup1 in microwave
14. Move-robot(above-table at-stove)
15. Put-in-microwave(cup1)
16. Close-door(microwave)
17. Heat-water-in-microwave(cup1) ; heat water
18. Open-door(microwave)
19. Pick-up-from-microwave(cup1)
20. Move-robot(above-table at-stove)

---

### Make Tea (Part 3)

21. Put-on-table(cup1) ; get a tea-bag and
22. Move-robot(above-table at-sink) ; put it in the cup
23. Pick-up-from-shelf(tea-box)
24. Move-robot(above-sink at-table)
25. Put-on-table(tea-box)
26. Open-container(tea-box)
27. Get-tea-bag
28. Make-tea(cup1)
29. Move-robot(above-table at-sink) ; dispose of it afterwards
30. Put-in-garbage-can(old-tea-bag)

---

### Planning Bias

- Practical planning (Kitchen domain)
  - 45 Objects, 50 Operators
  - 3.5 Branching factor, 30 Steps
- Must reduce search space
- Planning bias = Assumptions about
  - Structure of Plans (no knife)
  - Order of plans (simple -> complex)
Planning Strategy

- Method of exploiting a planning bias
- Several biases -> single strategy
  - Different methods for macros
  - Serial decomposability (Korf)
  - Peak to peak (Iba, James)
- Single bias -> several strategies
  - Plan order bias -> Case-based, Macro-based

Cases, Macros, Abstraction

- Comparison of three popular strategies
  - Search depth only
- Very different strategies
- Problem: Make a cup of instant coffee with sugar
- 42 Operators

Case-Based Planning

- Retrieve similar plan
- Adapt old plan to new situation
- Insert, remove, reorder, replace, change var.
  Binding, move current operator
- Adapt “Make-Tea” for Inst. Coff. W/ sugar
- Change var-binding, add ops, remove ops
- Add suffix plan to add sugar (14 steps)
Case-Based Example

22 Move-robot(at-table at-sink) ; identical to make-tea
23 Pick-up-from-shelf(instant-coffee-jar) ; replace tea-box with
24 Move-robot(at-sink at-table) ; instant-coffee-jar
25 Put-on-table(instant-coffee-jar)
26 Open-container(instant-coffee-jar)
27 Open-door(drawer) ; add steps to use spoon
28 Pick-up-from-shelf(spoon)
29 Scoop-instant-coffee
30 Pour-instant-coffee(1cup)
31 Stir(1cup) ; stir instant coffee \).
32 Put-down-on-table(spoon)

Case-Based Example

33 Move-robot(at-table at-sink) ; add sugar to instant coffee
34 Pick-up-from-shelf(sugar-box)
35 Move-robot(at-sink at-table)
36 Put-on-table(sugar-box)
37 Open-container(sugar-box)
38 Pick-up-from-table(spoon)
39 Scoop-sugar
40 Add-sugar(1cup)
41 Stir(1cup)
42 Put-on-table(spoon) ; Aaaa, remember that waiter ...
\textit{Jacques}... 

Macros

\begin{itemize}
  \item Compile short, general
  \item Reduce solution length
  \item Balance with increase
    \begin{itemize}
      \item Branching factor
      \item Matching cost
    \end{itemize}
  \item 10 Steps
  \item Example
    \begin{itemize}
      \item Put-In-Sink($Cup$)
      \item Fill-With-Water($Cup$)
      \item Turn-Water-Off
      \item Pick-Up-From-Sink($Cup$)
    \end{itemize}
\end{itemize}
Example: Macros

- DoLittle macro learner (23 macros)
- Minton: extract difficult subsequences
- Dynamic utility evaluation
- Learned macros (# 4 operators)
  - Fill a cup with water
  - Get a container (sugar, instant coffee, tea)
  - Scoop and stir

Abstraction-Based Planning

- Tyranny of details
- Ignore low level details
- Relaxed abstractions (Sacerdoti), reduced abstractions (Knoblock)
  - Example
    - Fetch cup
    - Fill Cup with water
    - Heat Cup
    - Add instant coffee
    - Add sugar
  - Fetch cup
    - open-door(cupboard)
    - pu-from-cupboard(cup1)
Example: Abstraction

- DoLittle abstraction learner (16 operators)
- Low level predicates removed
  - Position of robot
  - Container open
- Refinement of add sugar operator
  - 10 Steps
  - position, container at lower level

Comparison of search depths

- 42 Operators MEA
- Macros: 42/4 = 10
- Cases: Suffix plan is 14 steps
- Abstraction: Refinement search of add sugar is 10 ops

Motivation

- Use a case (Make-Tea) as first approx.
- Abstraction add-sugar
- Macros
  - Get a container
  - Add an ingredient
- Search depth 2
- Similar branching factor
Multi-strategy planning

- Reduce solution length
  - Break a problem into "subproblems"
  - Different or same abstraction level
  - Choose a planning strategy for each "subproblem"
  - Solve each subproblem
  - Combine solutions
- Interactions between strat. (EBL + cache)

Multi-strategy planning

- Uniform framework for strategies
  - Search through plan space
    - Plan language
    - Set of transformations
- Applicability conditions
  - Control branching factor
- Emulate search method

Plan Space Search Paradigm

- Plan space search (non-linear planning)
  - Nodes = partial plans
  - Edges = transformations on plans
- Planner is defined by
  - Plan language (totally/partially ordered)
  - Set of plan transformation
Example: Means Ends Analysis

Plan language:

Add operator to head:

Add operator to tail:

Advance current operator:

Example: Case-based planning

- Same plan language
- Plan transformations
  - Move active operator (forward/backward)
  - Insert, remove operator
  - Reorder operators
  - Change variable bindings
  - ...

Example: Reduced abstraction

- Plan language:
  - Similar to MEA (Plan head, plan tail)
  - Abstract search space
- Plan transformations:
  - MEA
  - Create abstract search space
  - Insert solution to abstract search space
DoLittle’s plan language

- Plan language:
  - Totally ordered
  - Instantiated variables
  - Plan head, plan tail
  - Trees of problem spaces
    - (abstract, serial, general)

DoLittle: Plan transformations

- Move active operator
- Insert operator sequence
- Remove, reorder, replace op. sequence
- Change variable binding
- Create problem space

DoLittle representation

- General operators:
  - STRIPS syntax
  - Very different semantics
- Representation of different strategies
  - Not just representation, but
    - Emulate effect on search space
    - e.g., macros vs. cases
General Operators

- When to apply a strategy?
  - Planner state description
    - Current world state
    - Set of open goals
  - Implicit representation
    - Explicit: Relevance lists (Prodigy)
    - Implicit: Similarity measure (Case-based)

Example: general operator

Operator{gen-pick-up-from-cupboard}
Variables Subject

- Preconds: (arm-empty) ; no subgoaling
  - (is-at robby at-table)
  - (is-in Subject cupboard)
- Open goals: (holding Subject) ; achieve this
- Effects: (holding Subject)
  - ~ (is-in Subject cupboard)
  - ~ (arm-empty)

What Strategy to Apply?

- Associated with a general op is a set of refinements
- Different refinement modes
  - Case - Abstract subgoal
  - Macro - Serial subgoal
  - Abstract op - Subgoal
Refinements

- Cupboard open or not?
  - Refine. 1
    - (pick-up-from-cupboard Subject)
  - Refine. 2
    - Abstract Subgoal
    - (pick-up-from-cupboard Subject)
- Refine. 2 Must not change any precond. literal (e.g., arm-empty)

Evaluation

- MEA, Case, Macro, Abstraction, PC-MSP-O Dolittle
- Two toy domains (Blocksworld, tohanoi)
- Kitchen domain
- No one single strategy planner superior
- Multi-strategy improved by a factor of 3.8
- Combination on a single problem

Time in the Kitchen Domain
Conclusion

- Uniform framework for strategies
  - plan language
  - set of transformations
- Representation for different strategies
  - General operators
- Combination of different planning strategies on a single problem improves performance